TECHNICAL RESEARCH REPORT

SYSTEMS RESEARCH CENTER

Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), Industry and the University

Generation of Alternative Feature-Based Models and Precedence Orderings for Machining Applications

by S.K. Gupta and D.S. Nau

TR 92-111

Generation of Alternative Feature-Based Models and Precedence Orderings for Machining Applications*

Satyandra K. Gupta† Dana S. Nau‡

University of Maryland

Abstract

For machining purposes, a part is often considered to be a *feature-based model* (FBM). i.e., a collection of machining features. However, often there can be several different FBMs of the same part. These models correspond to different sets of machining operations, with different precedence constraints. Which of these sets of machining operations is best depends on several factors, including dimensions, tolerances, surface finishes, availability of machine tools and cutting tools, fixturability, and optimization criteria. Thus, these alternatives should be generated and evaluated.

In this paper we present the following results:

1. We give general mathematical definitions of machining features and FBMs.

2. We present a systematic way to generate the alternative FBMs for a part, given an initial FBM for the part.

3. For each FBM, interactions among the features will impose precedence constraints on the possible orderings in which these features can be machined. We show how to generate these precedence constraints automatically for each interpretation.

4. We show how to organize the above precedence constraints into a time-order graph that represents all feasible orderings in which the features can be machined, and examine the time-order graph to see if it is consistent. If it is not consistent, then there is no way to machine this particular interpretation.

This work represents a step toward our overall approach of developing ways for automatically generating the alternative ways in which a part can be machined, and evaluating them to see how well they can do at creating the desired part. We anticipate that the information provided by this analysis will be useful both for process planning and concurrent design.

Keywords: feature-based modeling, geometric reasoning, manufacturing planning

*This work was supported in part by NSF Grant NSFD CDR-88003012 to the University of Maryland Systems Research Center and NSF grant IRI-8907890 to the University of Maryland Computer Science Department.

†Institute for Systems Research and Department of Mechanical Engineering. Email: skgupta@cs.umd.edu

‡Institute for Systems Research, Department of Computer Science, and Institute for Advanced Computer Studies. Email: nau@cs.umd.edu

1 Introduction

One of the missing links between CAD and CAM is the virtual absence of any systematic methodology for generating and evaluating the alternative ways to manufacture a proposed design [1]. Most integrated CAD/CAM systems try to generate a single process plan for a given design—but in general, there may be several alternative ways to manufacture the design, and these alternatives should be generated and examined, to determine how well each one balances the need for a quality product against the need for efficient manufacturing.

In this paper we will be focusing on machining operations. For the purposes of machining, the part is often considered as a collection of machinable features [2, 3], each of which can be created by one or more machining operations. To evaluate how well each machining step can do at creating the corresponding feature, we must take into account the feature geometry, tolerance requirements, surface finish requirements, and statistical variations in the process capabilities [4, 5, 6].

Given a part to be machined, there can be several different interpretations of the part as several different collections of machinable features [2, 7, 8]. Each such interpretation we call a *feature-based model* (FBM). Each sequence of machining operations capable of creating the part does so by creating the features in one of the FBMs for the part. Thus, one approach for generating the alternative machining sequences for a part is to generate the alternative FBMs, and for each FBM, to generate the various machining sequences capable of creating that FBM.

For generating and evaluating the machining alternatives for machined parts, we propose the following approach. Generate the alternative FBMs of the part, one by one. Each time an FBM is generated, generate the machining sequences that are capable of creating the features in that FBM, one by one. Each time a machining sequence is generated, evaluate it to see how good a job they can do of producing the part.

In this paper we present the following steps in the development of the proposed approach:

1. We give general mathematical definitions of machining features and FBMs.

2. We present a systematic way to generate the alternative FBMs for a part, given an initial FBM for the part.

3. For each FBM, interactions among the features will impose precedence constraints on the possible orderings in which these features can be machined. We show how to generate these precedence constraints automatically for each interpretation.

4. We show how to organize the above precedence constraints into a time-order graph that represents all feasible orderings in which the features can be machined, and examine the time-order graph to see if it is consistent. If it is not consistent, then there is no way to machine this particular interpretation.

This paper is organized as follows. Section 2 discusses related work. Section 3 presents our definitions. Section 4 presents our algorithm for generating precedence constraints and time-order-graphs from FBMs. Section 5 presents our algorithm for generating alternative FBMs for a part. Section 6 contains concluding remarks.

2 Review of Previous Research Work

Because of the recent popularity of feature based approaches in variety of CAD/CAM implementations, a vast amount of literature attempts to define term *feature(s)* [3, 9, 10, 11, 12, 13, 14]. Researchers in different domains use the term *feature* to mean different things. Significant amount of research efforts have been directed towards defining a set of form features which will serve as communication medium between the design and the manufacturing domains. But at present, most of the researchers are convinced that a single set of features cannot satisfy the requirements of both the domains. Recent trend seems toward defining a set of features which are suitable for a specific application domain such as machining, assembly, inspection etc. Most of the researchers agree that volumetric features are preferable over surface features for the purposes of machining.

There are three primary approaches for obtaining features from a CAD model. In *human-supervised feature recognition*, a human user examines an existing CAD model to determine what the manufacturing features are [15]. In *automatic feature recognition*, the same feature recognition task is performed by a computer system [7, 16, 17, 18]. In *design by features*, the designer specifies the initial CAD model in terms of various form features which translate directly into the relevant manufacturing features [3, 19, 20]. However, each of these approaches typically produces a single set of features describing the CAD model, rather than several alternative interpretations of the model.

Hummel [2] and Mantyla [21] present examples of multiple representation of the same object for the different functional domains such as design and machining. These papers also present some examples of multiple feature interpretations for same part in the machining domain. However, these papers do not describe a system or methodology for generating multiple feature models.

Hayes [22, 23] has built a program called Machinist, which captures some of the knowledge used by machinists during process planning. Hayes defines an interaction to occur among two features when one of the results of making one feature is to destroy one of the preconditions required in order to make another feature. This program has the capability to identify certain kinds of feature interactions that affect how an object should be machined. However, its representation of features is not adequate for all aspects process planning. For example, if the Machinist program decides that some hole needs to be made before some slot, it does not automatically update the dimensions of the hole or the slot—information which would be needed to select what machining machining processes to use for these features.

The AMPS process planning system [24] includes a step called "feature refinement," which consists of heuristic techniques for combining a set of features into a more complex feature if it appears that this will optimize the plan, or splitting a feature that cannot be machined into two or more features that can (hopefully) be machined. Since the techniques are heuristic in nature, it is not entirely clear when alternative interpretations will be produced.

Vandenbrande [7] has developed a system that combines techniques from artificial intelligence and solid modeling. The program uses hints or clues to identify potential features in the boundary representation of the part. The system is capable of identifying interacting features (e.g., two intersecting slots). This program also produces alternative feature interpretations in certain cases. But since there is no formalization available regarding the kinds of interactions it handles, it is hard to determine what all the interpretations it produces are.

The first systematic work in the direction of generation of alternative interpretations was done by Karinthi and Nau [8, 25]. He describes an approach for producing alternative interpretations of the same object as different collections of machining features as the result of algebraic operations on the features, and a system for generating alternative interpretations by performing these algebraic operations. This system works with abstract volumetric features. There is no direct relation between these features and machining operations. Therefore some of the interpretations generated by this approach are not feasible from the machining point of view. In this approach a set of algebraic operators (such as maximal extension, truncation etc.) has been used to generate new interpretations of the part. But this set of operators is not sufficient to generate all interpretations of the part. Moreover, many times the resulting features do not belong to any of the feature classes. Some of the feature interactions may also result in partial ordering among features, which is an important issue from a machining point of view—but this work does not deal with time orderings among the features.

3 Background

3.1 Preliminary Definitions

For our purposes, a *solid* is any regular, semi-analytic subset of three-dimensional Euclidean space. If R is any solid, then $b(R)$ is the *boundary* of R, and $\iota(R)$ is the *interior* of R. Note that $R = \iota(R) \cup b(R)$ and that $\iota(R) \cap b(R) = \emptyset$. A *patch* of R is a regular, semi-analytic subset of the boundary $b(R)$. If R and R' are solids, then $R \cap^* R'$ is the *regularized intersection* of a and b, i.e., the closure of $\iota(R) \cap \iota(R')$. Similarly, $R \cup^* R'$ and $R -^* R'$ are the *regularized union* and *regularized difference*, respectively.

A *machined part* (or just a *part*) is the finished component to be produced as a result of a set of machining operations on a piece of *stock*, i.e., the raw material from which the part is to be machined. We will represent both the part and the stock as geometric solids.

Throughout this paper, we let P be a solid representing a part, and S be a solid representing the stock from which P is to be made. The *delta volume* (i.e., the volume to be machined), is the solid $\Delta = S - P$.

3.2 Machining Features

To perform a machining operation, one starts out with a rotating cutting tool. The cutting tool is mounted on a large machine tool, and the total volume occupied by the cutting tool and the machine tool is quite large. But we will only be interested in some small portion of this total volume, namely the portion that actually gets close to the workpiece. We will call this portion the *tool volume*, and we will denote it by T. The boundary $b(T)$ is naturally partitioned into three patches, as shown in Fig. 1(a):

- the *separation patch* $s(T)$, i.e., the portion of $b(T)$ that connects to the rest of the machine tool;
- the *cutting patch* $c(T)$, i.e., the portion of $b(T)$ that is capable of cutting metal;
- the *non-cutting patch* $n(T)$, i.e., the portion of $b(T)$ that is not capable of cutting metal.

For the purpose of locating the tool, we will choose a particular point p_{td} of T as a *datum point.* Usually p_{td} will be the tip of the cutting tool, but not always.

In an abstract sense, a machining feature has something to do with the volume of material removed by a machining operation. However, in order to define what we mean by a feature, we will need to know not just the volume of material which the feature can remove from the workpiece, but also what kind of machining operation we are performing, and how we access the workpiece in order to perform the operation. Thus, a feature will be a triple

$$f = (rem(f), acc(f), class(f)),$$

where $rem(f)$, $acc(f)$, and $class(f)$ are as defined below.

- To perform the machining operation, one sweeps the tool volume T along some trajectory t. Given a tool T and a workpiece W, the trajectory t is *feasible* for T and W only if sweeping T along t does not cause interference problems between the non-cutting surface $n(T)$ and the workpiece. Fig. 1(b) shows an example of a feasible tool trajectory for drilling.

  If t is feasible, then the volume created by sweeping T is

  $$T_{sw} = \{(p - p_{td}) + q : p \in T \text{ and } q \in t\},$$

  as shown in Fig. 1(c). However, only a portion of T_{sw} actually corresponds to our machining feature. In particular, let *approach plane* π be the plane perpendicular to t at the point p_{td}, as shown in Fig. 1(a). Then $rem(f)$, the *removal volume* of f, is the solid consisting of all points in T_{sw} that are on or below π.

- The *accessibility volume* for f is the set $acc(f)$ of all points in T_{sw} that are on or above π. The *approach patch* $a(f)$ is that portion of $b(rem(f))$ consisting of all points in $rem(f) \cap \pi$.

- The feature f will be an *instance* of some *feature class* ϕ, which is a parameterized set of machining features characterized by the shape and trajectory of the cutting tool. If f is a feature in ϕ, then the *class* of f is $class(f) = \phi$. If f is an instance of ϕ, then the f's *parameters* in ϕ are the specific set of parameter values for ϕ that yield f. Below are two examples:

  - If we are interested in drilling holes, then we may define ϕ_h to be the set of all features that can be created by sweeping a drill bit of diameter d along a linear trajectory starting at the datum point p_{td} and going in some direction $\vec{v}$ for some distance l. Thus, we can specify a particular feature in ϕ_h by giving specific values for d, p_{td}, $\vec{v}$, and l.

  - If we are interested in making rectangular pockets or slots, then we may define ϕ_p to be the set of all features that can be created by sweeping an end mill of radius r along the trajectory shown in Fig. 2(b), whose parameters are the starting point p_{td}, the depth d, the length l, the width w, and the orientation vector $\vec{v}$. Thus, we can specify a particular feature in ϕ_p by giving specific values for r, p_{td}, d, l, w, and $\vec{v}$.

  Fig. 1(d) and Fig. 2(c) show some examples of the feature types.

3.3 Feature-Based Model

Normally we will have some fixed finite set of feature classes

$$\Phi = \{\phi_1, \ldots, \phi_n\},$$

and for each part that we want to manufacture, we will be interested in describing the part in terms of features from Φ. Each set of features from Φ that describes the part is a feature-based model of the part. This is made formal below.

Suppose we are given a part P and stock S. A *feature-based model* (or FBM) of P and S is any set of features F having the following properties:

1. Each $f \in F$ is an instance of some feature class in Φ.
2. If we subtract the features in F from S, we get P; i.e., $S - \bigcup_{f \in F}(rem(f)) = P$.
3. No feature in F is redundant, i.e., for every feature $f \in F$, $S - \bigcup_{g \in F-\{f\}}(rem(g)) \neq P$.

Intuitively, an FBM is an interpretation of the delta volume as a set of machining features. For example, the set $\{s11, s12, s13, h10\}$ shown in Fig. 4(a) is an FBM of the part and stock shown in Fig. 3. Furthermore, for a given part P and stock S, there may be more than one FBM. For example, Fig. 4 shows four different FBMs for the same part and stock.

Two FBM's F and F' are *equivalent* if they represent the same part and stock. In case of two equivalent FBMs F and F', we can say that F' is a *reinterpretation* of F or vice versa. For example, in Fig. 4, all models are reinterpretations of each other.

Let f and f' be any two distinct features in some FBM. Then f and f' *intersect* each other if $rem(f) \cap^* rem(f') \neq \emptyset$. Furthermore, f and f' are *adjacent* if $rem(f) \cap rem(f') \neq \emptyset$ and $rem(f) \cap^* rem(f') = \emptyset$.

4 Precedence Constraints in Feature-Based Models

Due to accessibility [6], setup [22] and other types of interactions among the features in a given FBM, features can not be machined in any arbitrary order. Instead, these interactions will introduce *precedence constraints* requiring that some features be machined before or after other features.

Let F be an FBM, and let f and f' be any two features in F. We will be interested in the following two types of precedence constraints among f and f':

1. *Accessibility precedence constraint:* If the cutting-tool approaches f through the volume occupied by f', then f' must be machined before f. An example is shown in Fig. 5(b). Pocket p need be machined before machining hole h.

   In some cases, it may be possible to machine f if either one of two features f', f'' are machined first. For example, let us assume that rectangular pocket and hole are the only available feature classes. It will be possible to machine hole h (as shown in Fig. 6) after machining either of the pockets $p1$ or $p2$. We specifically exclude such cases in this paper, but we intend to address such cases in future work. However, one should notice that such cases usually do not arise in practice. In a practical situation any restriction on the shapes of the pockets will be quite artificial. In most of the situations both of these pockets will be merged into a more complex single pocket.

2. *Feature minimality precedence constraint:* Suppose that machining f' before f would create a situation in which it will be possible to machine $rem(f)$ using a smaller feature g of the same class as f. Then f must machined before f'. An example is shown in Fig. 5(c). If pocket p is machined before machining hole h', it will be possible to machine $rem(h')$ by a smaller feature h.

Given an FBM F, a *machining order* for F is any total ordering $\{f_1, f_2, \ldots, f_k\}$ satisfying the precedence constraints defined above. More specifically, a machining order for F is any total ordering $\{f_1, f_2, \ldots, f_k\}$ such that if we let $W_0 = S$ and $W_i = W_{i-1} - f_i$ for all $i > 0$, then the following conditions are satisfied:

- for all $i > 0$, f_i is accessible in W_{i-1}, i.e., $acc(f_i) \cap^* W_{i-1} = \emptyset$.
- each f_i is the smallest feature in its class that can be used to produce W_i from W_{i-1}; i.e., there is no feature $f \in class(f_i)$ such that $rem(f) \subset rem(f_i)$ and $W_{i-1} - rem(f_i) = W_{i-1} - rem(f)$.

If there is no machining order for F, then this means that the precedence constraints contradict each other, so that it is not possible to machine F. If there is at least one machining order for F, then we say that F is *machinable.*

4.1 Creating a Time-Order Graph

Given an FBM F, we can determine the precedence constraints for F by examining the features in f. These precedence constraints define a partial order $\prec$ on the features in F. This partial order represents all possible time orderings in which the features might be machined, in the sense that every total ordering consistent with $\prec$ is a machining order for F.

We define the *time-order graph* for F to be the digraph $G_{prec} = (F, E_{prec})$, where the edge set E_{prec} contains an edge from each feature to its immediate successors.[1]

Given an FBM F, we want to examine the features in F to determine whether F is machinable, and if so, what its time-order graph is. The following procedure will do this.

procedure CREATE-TIME-ORDER-GRAPH(F)

Step 1. Find all accessibility precedence constraints in F, using the FIND-ACCESSIBILITY-CONSTRAINTS procedure described below. If this procedure exits with failure, then F is not machinable. Otherwise, it will return a digraph (F, E) such that E is a subset of the set E_{prec} that we want to find.

Step 2. Find all feature minimality precedence constraints in F, using the FIND-FEATURE-MINIMALITY-CONSTRAINTS procedure described below. If FIND-FEATURE-MINIMALITY-CONSTRAINTS exits with failure, then F is not machinable. Otherwise, it will augment (F, E) to produce the desired digraph G_{prec}.

Fig. 7(a) and Fig. 7(b) shows an example part and its time-order graph.

[1] This graph, which is called the *Hasse diagram* for $\prec$, is a standard representation of a partially ordered set (e.g., see [26]).

4.2 Finding Accessibility Precedence Constraints

Let F be an FBM, and f be any feature in F. If F is machinable, then an *enabling set* for f and the associated *enabling volume* are defined recursively as follows:

1. if f is accessible in S, then $\emptyset$ is an enabling set for f, and the associated enabling volume is $\emptyset$.

2. Suppose $Y = \{g_1, \ldots, g_k\}$ is a set of features. Suppose that for each i, Y_i is an enabling set for g_i, and V_i is the associated enabling volume. If f is accessible in the solid $W = S - (V_1 \cup \ldots \cup V_k) - (rem(g_1) \cup \ldots \cup rem(g_k))$, then Y is an enabling set for f, and $V = S - W$ is the associated enabling volume.

Intuitively, if Y is an enabling set for f, then once the associated enabling volume V has been machined, it will be possible to machine f.

Let f be a feature, and Y be an enabling set for f. Then Y is *minimal* if no subset of Y is an enabling set for f. Since we are excluding the type of situation shown in Fig. 6, it follows that the enabling volume for each feature f in F is unique. If F is a machinable FBM, then the following algorithm will find the minimal enabling set for each feature in F.

procedure FIND-ACCESSIBILITY-CONSTRAINTS

1. Set $FINISHED := \emptyset$, and $REMAINING := F$.

2. while $REMAINING \neq \emptyset$ do

   (a) Set $CURRENT :=$ the set of all features in $REMAINING$ that are accessible in $S - \bigcup_{h \in FINISHED}(rem(h))$.

   (b) If $CURRENT = \emptyset$, then F is not a machinable FBM, so exit with failure.

   (c) For each feature $f \in CURRENT$, do the following:

      i. $X := \{$all features in FINISHED that intersect $acc(f)\}$.

      ii. By enumerating the subsets of X in order of least cardinality first, find the smallest subset $Y = \{g_1, \ldots, g_k\}$ of X such that f is accessible in the solid $W = S - (V(g_1) \cup \ldots \cup V(g_k)) - (rem(g_1) \cup \ldots \cup rem(g_k))$.

      iii. Y and $S - W$ are f's minimal enabling set and associated enabling volume, so set $e(f) := Y$ and $V(f) := S - W$.

   (d) Set $FINISHED := FINISHED \cup CURRENT$, and
   $REMAINING := REMAINING - CURRENT$.

3. Return the set $\mathcal{V} = \{V(f) : f \in F\}$ and the graph (F, E), where E is the set of all ordered pairs (f_1, f_2) such that $f_2 \in F$ and $f_1 \in e(f_2)$.

4.3 Finding Feature Minimality Precedence Constraints

procedure FIND-FEATURE-MINIMALITY-CONSTRAINTS$(F, E, \mathcal{V})$

1. For every intersecting pair of features (f, f') in F, do the following:

(a) If there is a feature $g \in class(f)$ such that $rem(g) \subset rem(f)$ and $S - V(f') - rem(f') - rem(f) = S - V(f') - rem(f') - rem(g)$, then insert the edge (f, f') into E.

(b) If there is a feature $g \in class(f')$ such that $rem(g) \subset rem(f')$ and $S - V(f) - rem(f) - rem(f') = S - V(f) - rem(f) - rem(g)$, then insert the edge (f', f) into E.

2. If G_{prec} is cyclic then exit with failure. Otherwise, return the digraph (F, E).

5 Generating Equivalent Feature-Based Models

If we are given an initial FBM F for some part P and stock S, then producing other interpretations equivalent to F can be produced by manipulating the features in F. For example, Fig. 4(b) can be produced from Fig. 4(a) by splitting the hole h10, and Fig. 4(c) can be produced from Fig. 4(a) by reorienting the shoulders s11 and s12.

Given F, we will want not only to generate other interpretations equivalent to F, but also to generate their time-order graphs. One approach for this task would be to define feature manipulation operators that produce alternative FBMs, and then use the CREATE-TIME-ORDER-GRAPH algorithm of Section 4 to produce the associated time-order graphs. However, such an approach would be very expensive computationally, because of the repeated calls to CREATE-TIME-ORDER-GRAPH.

A much more efficient approach can be devised by noting that given an FBM F and its time-order graph G_{prec}, if we apply a feature manipulation operator to produce an alternative interpretation F', we can produce the associated time-order graph G'_{prec} at the same time, by making some simple changes to G_{prec}. This is how we define our feature manipulation operators below.

5.1 Definitions of Feature Manipulation Operators

In this section we define the feature manipulation operators. Given an FBM F and its time-order graph G_{prec}, each operator generates an alternative FBM F' along with the associated time-order graph G'_{prec}.

We describe ENLARGE(f, g) operator in some detail. For the sake of brevity, we omit the details of other operator; these are similar to the ENLARGE(f, g) operator.

Enlarging a Feature. Given a feature f and a neighboring feature g, the ENLARGE(f, g) operator tries to produce a feature f' that subsumes f, by extending f into g. If it finds such a feature, it returns the resulting FBM and its time-order graph. We represent feature classes as parameterized solids. Therefore, existence of a larger or smaller features can be examined by manipulating the various parameters of the feature class. Following procedure illustrates a portion of the ENLARGE(f, g) operator. In this procedure we enlarge a feature f with respect to its immediate predecessor g.

procedure ENLARGE(f, g)

1. If $(g, f) \notin E_{prec}$ or $acc(f) \cap^* rem(g) = \emptyset$ then exit with failure

2. Try to find enlarged feature f' such that

   (a) $f' \in class(f), rem(f') \supset rem(f), acc(f') \cap^* acc(f) \neq \emptyset$
   (b) $rem(f') \cap^* P = \emptyset$
   (c) $acc(f') \cap^* rem(g) = \emptyset$
   (d) f' is minimal, i.e., $\nexists h$ such that $h \in class(f'), rem(h) \subset rem(f')$

   If no such f' is found then exit with failure.
   Otherwise continue as follows:

   (a) Find a feature g' such that
      i. $g' \in class(g)$
      ii. $rem(g') \supseteq rem(g) - rem(f')$
      iii. g' is minimal, i.e., $\nexists h$ such that $h \in class(g'), rem(h) \subset rem(g')$
   (b) Create F' and G'_{prec} as follows:
      i. $F' = (F - \{f, g\}) \cup \{f',g'\}$
      ii. $E'_{prec} = E_{prec} - (g, f)$
      iii. if $g' = g$ then insert (f, g) in E'_{prec}.
      iv. Let Y be the set of predecessors of g, let Y' be the subset of Y such that $Y' = \{y \mid y \in Y, acc(f') \cap^* y \neq \emptyset\}$, $\forall y \in Y'$ insert (y, f) in E'_{prec}

3. return F' and G'_{prec}

Let us consider Fig. 5(a) that shows a part with two features a pocket, and a hole at the base of the pocket. From a machining perspective, the part can be characterized by the features p and h (shown in Fig. 5(b)) or, the features p and h' (shown in Fig. 5(c)), where h' is an extension of h all the way to the top of the pocket. Which of the feature sets, $\{p, h\}$ or $\{p, h'\}$ is preferable depends on machining considerations. We can get h' by enlarging h with respect to p.

Reducing a Feature. Given a feature f and an intersecting feature g, the REDUCE(f, g) operator tries to produce a feature f' subsumed by f, by truncating f at g. If it finds such a feature, it returns the resulting FBM and its time-order graph.

Reorienting a Feature. Given a feature f, the REORIENT(f) operator tries to find a feature f' of the same class in some other orientation. This operator may find more then one reoriented feature f'. If it finds such a feature(s), it returns the resulting FBM(s) and its time-order graph(s).

Splitting a Feature. Given a feature f and an intersecting feature g, the SPLIT(f, g) operator tries to split feature f into two or more features. If it finds a valid split, it returns the resulting FBM and its time-order graph.

Combining Features. Given a feature f and a set of neighboring features X of feature f, the COMBINE(f, X) operator tries to combine the members of set X into single feature f'. This operator may find more then one combined feature f'. If it finds a combined feature(s), it returns the resulting FBM(s) and its time-order graph(s).

As an example, suppose we start with Interpretation 1 of Fig. 8. Then Interpretation 2 can be generated by enlarging h with respect to $s1$, Interpretation 3 can be generated by reorienting f, and Interpretation 4 can be generated by splitting h with respect to $s3$.

5.2 Algorithm

By starting with a given FBM M of P and S and performing successive reinterpretations, it is possible to produce other FBM's of P and S. The following state-space search algorithm[2] will generate all FBMs that can be produced in this manner.

procedure FIND-MODELS(F)

1. Set $OPEN := F$, and $CLOSED = \emptyset$
2. While $OPEN \neq \emptyset$, do the following:
   (a) Choose some model $F \in OPEN$. For each $f \in F$ do
      i. Let X be the set of feature which are adjacent to f, and Y be the set of features that intersect f. Let $Z = X \cup Y$.
      ii. Let M be the set of new FBM's that can be found by applying feature manipulation operators to f; i.e.,

$$\begin{aligned} \mathrm{M} &= \text{REORIENT}(f) \cup \{\text{ENLARGE}(f,g) : g \in Z\} \\ &\quad \cup \{\text{REDUCE}(f,g) : g \in Y\} \\ &\quad \cup \{\text{SPLIT}(f,g) : g \in Y\} \\ &\quad \cup \{\text{COMBINE}(f,Z)\}. \end{aligned}$$

      iii. For each model $M \in$ M that is not in $CLOSED \cup OPEN$, insert M into $OPEN$.
      iv. Remove F from $OPEN$ and insert it into $CLOSED$
3. return $CLOSED$

As an example, Fig. 9 shows a portion of the state space that would be generated by this algorithm, starting from any one of the states shown in Fig. 8. The reason why Fig. 9 is only a portion of the complete state space is that it shows only those states that result from applying feature manipulation operators to the holes. If we also apply feature manipulation operators to the shoulders (for example, see Fig. 4), we will obtain the complete state space, which contains 32 nodes (four nodes for each node shown in Fig. 9).

[2] For more details on state-space search algorithms, see [27].

6 Conclusions

In this paper, we have given general mathematical definitions of machining features and FBMs, and presented a systematic way to generate the alternative FBMs for a part, given an initial FBM for the part. We have shown how to generate precedence constraints for the features in an FBM, and organize these features into a time-order graph representing the feasible orderings in which the features can be machined.

This work represents a step toward our long-term goal of developing ways for automatically generating the alternative ways in which a part can be machined, and evaluating them to see how well they can do at creating the desired part. As another step toward this goal, we have also developed ways to evaluate how well a machining process can do at creating a machined feature; our work on this topic is described in [1, 6, 28].

Some of the benefits of this approach are listed below:

1. *Pushing process engineering upstream:* By using domain-specific features, we will be incorporating process-related information in the features themselves. This allows an easy mapping of machining features to machining operations.

2. *A sound theoretical basis:* As opposed to existing rule based approaches, our approach is based on theoretical foundations, which we hope will enable us to make rigorous statements about the soundness, completeness, efficiency, and robustness of the approach.

3. *Focus on alternatives:* The information provided by this research will enable us to provide the necessary information to a manufacturing engineer or a process planning system about the alternative ways in which the part might be machined. This will help in developing alternative process plans. Depending upon machine tool availability and/or other constraints specific to plant facilities, one can choose a appropriate process plan. The results of machinability evaluation can be used to provide feedback to the designer about the machinability of the design, so the designer can modify the design if necessary to balance the need for efficient machining against the need for a quality product.

We anticipate that the results of this work will be useful in providing a way to speed up the evaluation of new product designs in order to decide how or whether to manufacture them. Such a capability will be especially useful in flexible manufacturing systems, which need to respond quickly to changing demands and opportunities in the marketplace.

References

[1] D. S. Nau, G. M. Zhang, S. K. Gupta, and R. R. Karinthi. Evaluating product machinability for concurrent engineering. In W. G. Sullivan and H. R. Parsaei, editors, *Handbook of Concurrent Design and Manufacturing.* Chapman and Hall, 1992. To appear, also available as SRC Tech Report TR 92-29.

[2] K. E. Hummel. The role of features in computer-aided process planning. In *Proceedings of Feature Symposium*, Woburn, Boston, MA, August 1990.

[3] J. J. Shah. Philosophical development of form feature concept. In *Proceedings of Feature Symposium*, Woburn, Boston, MA, August 1990.

[4] G. M. Zhang and S. G. Kapoor. Dynamic generation of machined surface, part- i: Mathematical description of the random excitation system. *Journal of Engineering for Industry, Transaction of ASME*, May 1991.

[5] G. M. Zhang and S. G. Kapoor. Dynamic generation of machined surface, part- ii: Mathematical description of the tool vibratory motion and construction of surface topography. *the Journal of Engineering for Industry, Transaction of ASME*, May 1991.

[6] D. S. Nau, G. M. Zhang, and S. K. Gupta. Generation and evaluation of alternative operation sequences. In *Accepted in ASME Winter Annual Meeting*, 1992. Also available as SRC Tech Report TR 92-20.

[7] Jan H. Vandenbrande. *Automatic recognition of machinable features in solid models.* PhD thesis, Electrical Engineering Department, University of Rochester, 1990.

[8] R. R. Karinthi and D. S. Nau. An algebraic approach to feature interactions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(4):469–484, April 1992.

[9] K.E.Hummel and S.L. Brooks. Symbolic representation of manufacturing features for an automated process planning system. Technical Report Technical Report BDX-613-3580, Bendix Kansas City Division, 1986.

[10] A. L. Clark and N. K. South. Feature based design of machined parts. In *Proccedings of AUTOFACT87*, Detroit,MI, November 1987.

[11] A. Klien. A solid groove feature based programming of parts. *Mechanical Engineering*, March 1988.

[12] S. Drake and S. Sela. A foundation for features. *Mechanical Engineering*, January 1989.

[13] N.N.Z. Gindy. A hierarchical structure for form features. *Int. J. Prod. Res.*, 27(12):2089–2103, 1989.

[14] D. Genord. Features: Enablers of automation. In *Proceedings of Feature Symposium*, Woburn, Boston, MA, August 1990.

[15] P. Brown and S. Ray. Research issues in process planning at the national bureau of standards. In *19th CIRP International Seminar on Manufacturing Systems*, pages 111–119, June 1987.

[16] Hiroshi Sakurai and David C. Gossard. Recognizing shape features in solid models. *IEEE Computer Graphics & Applications*, September 1990.

[17] Xin Dong. *Geometric feature extraction for computer aided process planning.* PhD thesis, Rensselaer Design Center, Rensselaer Polytechnic Institute, Troy, New York, 1988.

[18] Sanjay Joshi. *CAD interface for automated process planning.* PhD thesis, Purdue University, 1987.

[19] G. P. Turner and D. C. Anderson. An object oriented approach to interactive, feature based design for quick turn around manufacturing. In *ASME-computers in Engineering Conference*, San Fransisco, CA, July-Aug. 1988.

[20] N.C. Ide. Integration of process planning and solid modeling through design by features. Master's thesis, University of Maryland, College Park, Department of Computer Science, 1987.

[21] M. Mantyla, J. Opas, and J. Puhakka. Generative process planning of prismatic parts by feature relaxation. Technical report, Helsinki Institute of Technology, Laboratory of Information Processing Science, Finland, Feb 1989.

[22] C. C. Hayes and P. Wright. Automatic process planning: using feature interaction to guide search. *Journal of Manufacturing Systems*, 8(1):1–15, 1989.

[23] C. C. Hayes. *Machine Planning: A Model of an Expert Level Planning Process.* PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1990.

[24] Tien-Chien Chang. *Expert Process Planning for Manufacturing.* Addison-Wesley Publishing Co., 1990.

[25] R. Karinthi, D. Nau, and Q. Yang. Handling feature interactions in process planning. *Applied Artificial Intelligence*, 1992. Special issue on AI for manufacturing. To appear.

[26] F. P. Preparata and R. T. Yeh. *Introduction to Discrete Structures.* Addison-Wesley, Reading, MA, 1973.

[27] Nils Nilsson. *Principles of Artificial Intelligence.* Morgan Kaufmann Publishers Incorporated, CA, 1980.

[28] Dana S. Nau, Guangming Zhang, and Satyandra Gupta. Generation of machining alternatives for machinability evaluation. In *NSF Design and Manufacturing Systems Grantees Conference*, UNCC, Charlotte, NC, January 1993. To appear.

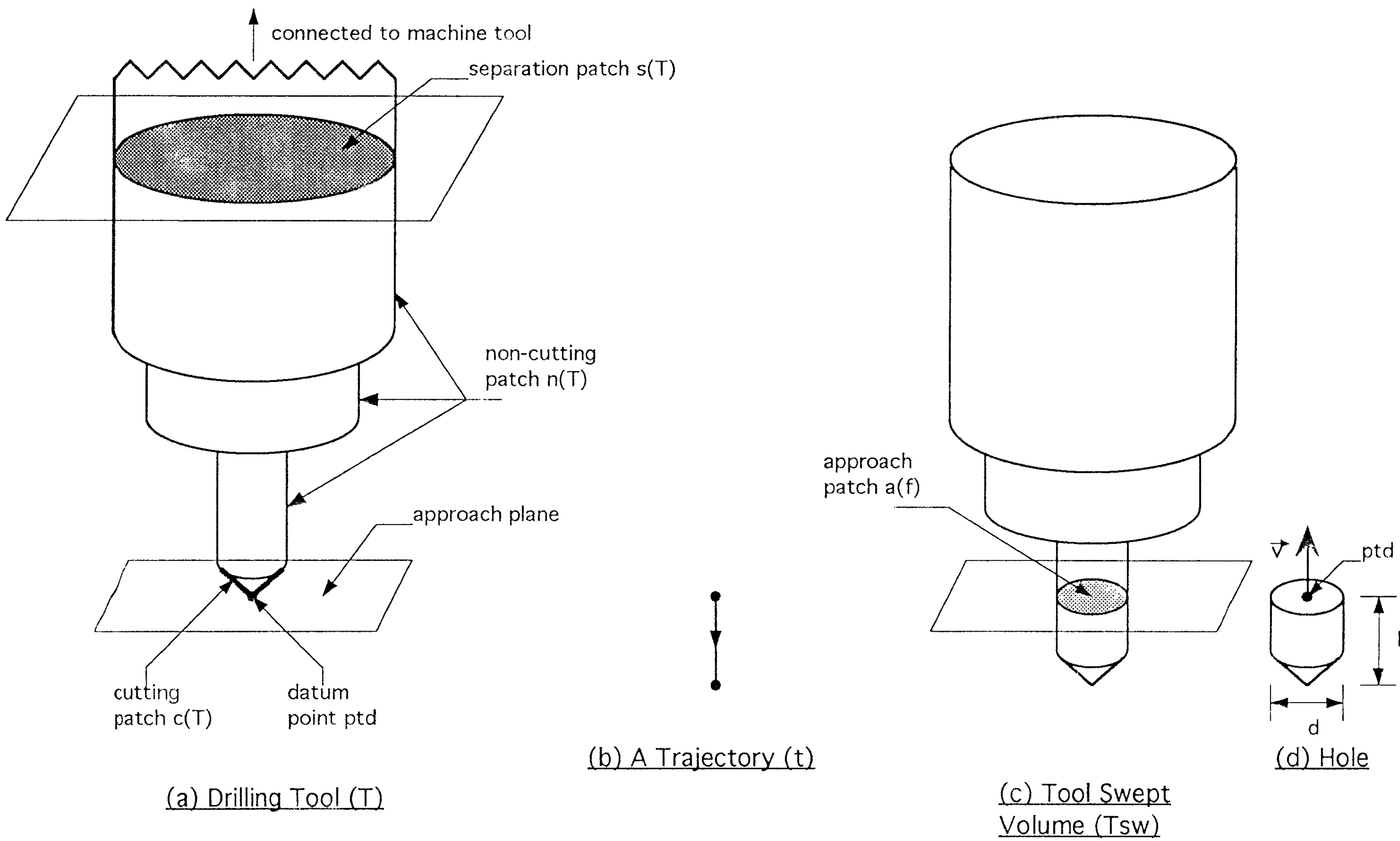
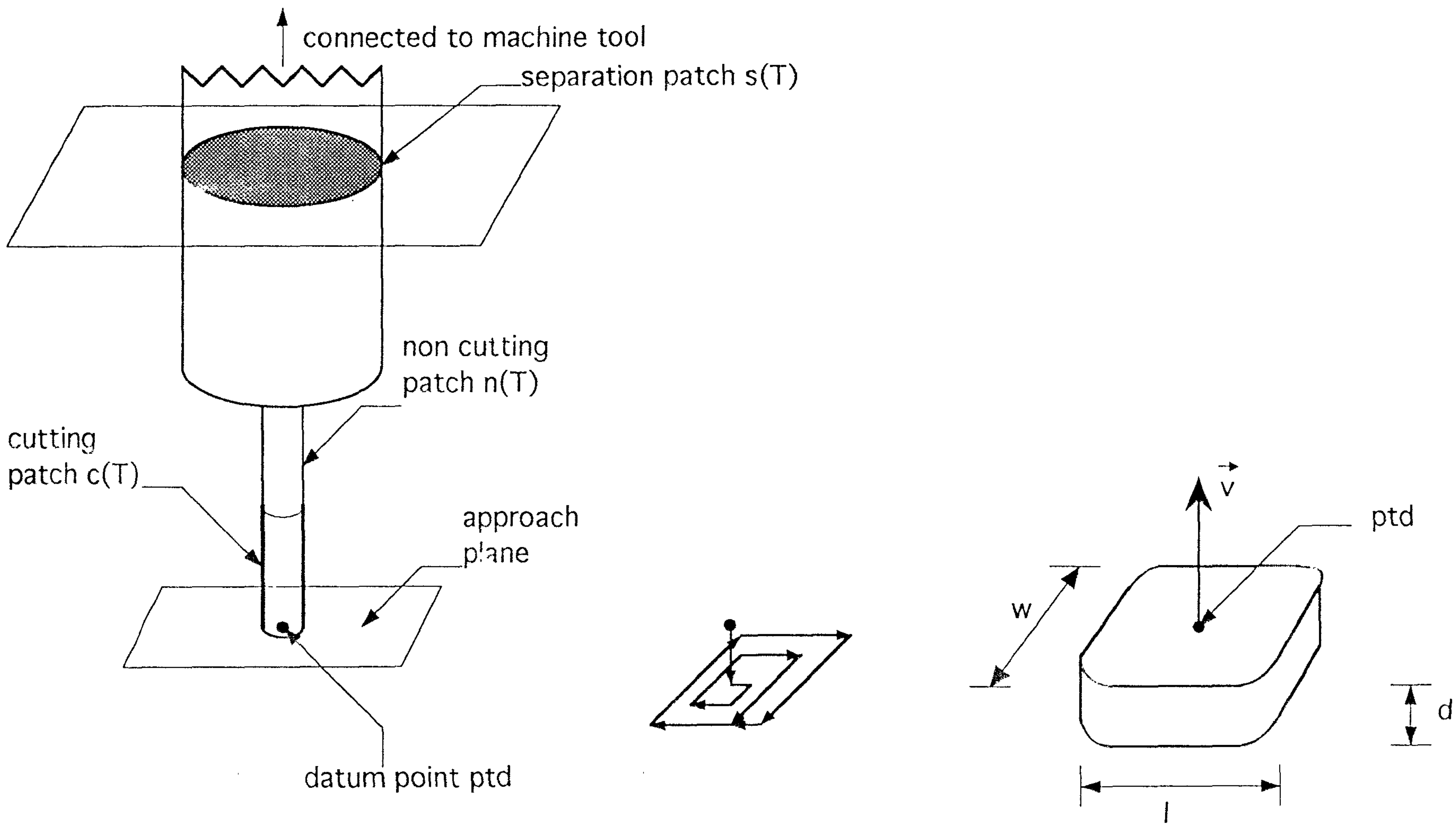

connected to machine tool
separation patch s(T)
non-cutting
patch n(T)
approach plane
cutting
patch c(T)
datum
point ptd
(a) Drilling Tool (T)
(b) A Trajectory (t)
approach
patch a(f)
(c) Tool Swept
Volume (Tsw)
ptd
d
l
(d) Hole

Fig. - 1

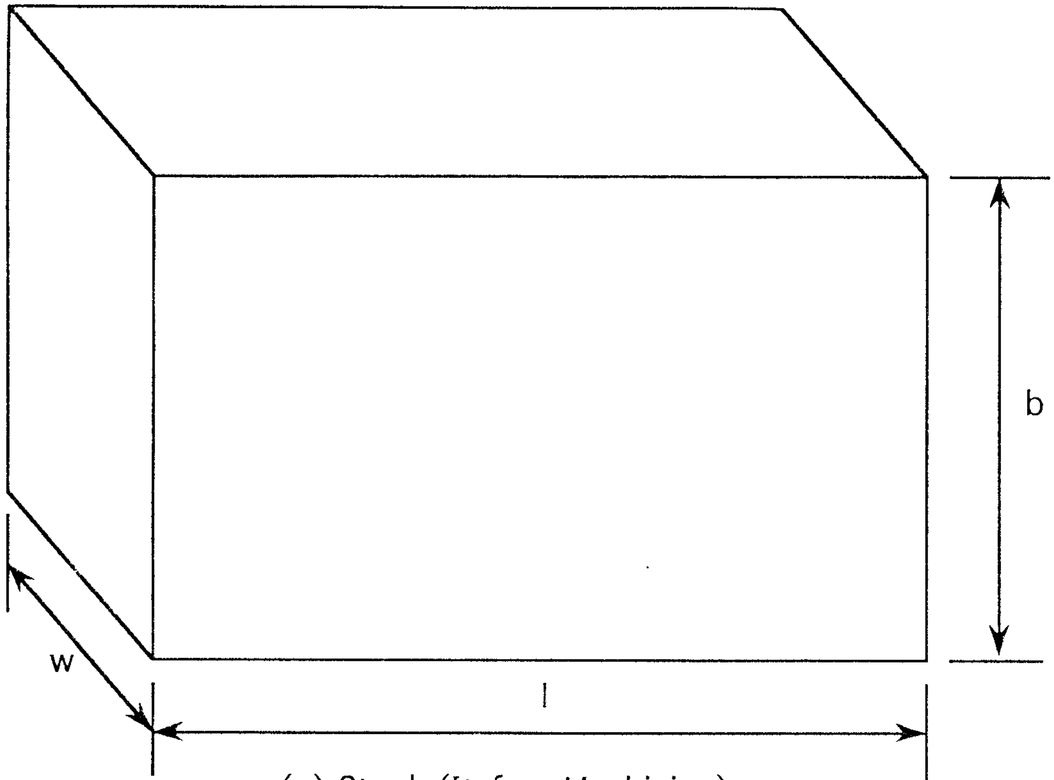


(a) End Milling Tool (T)

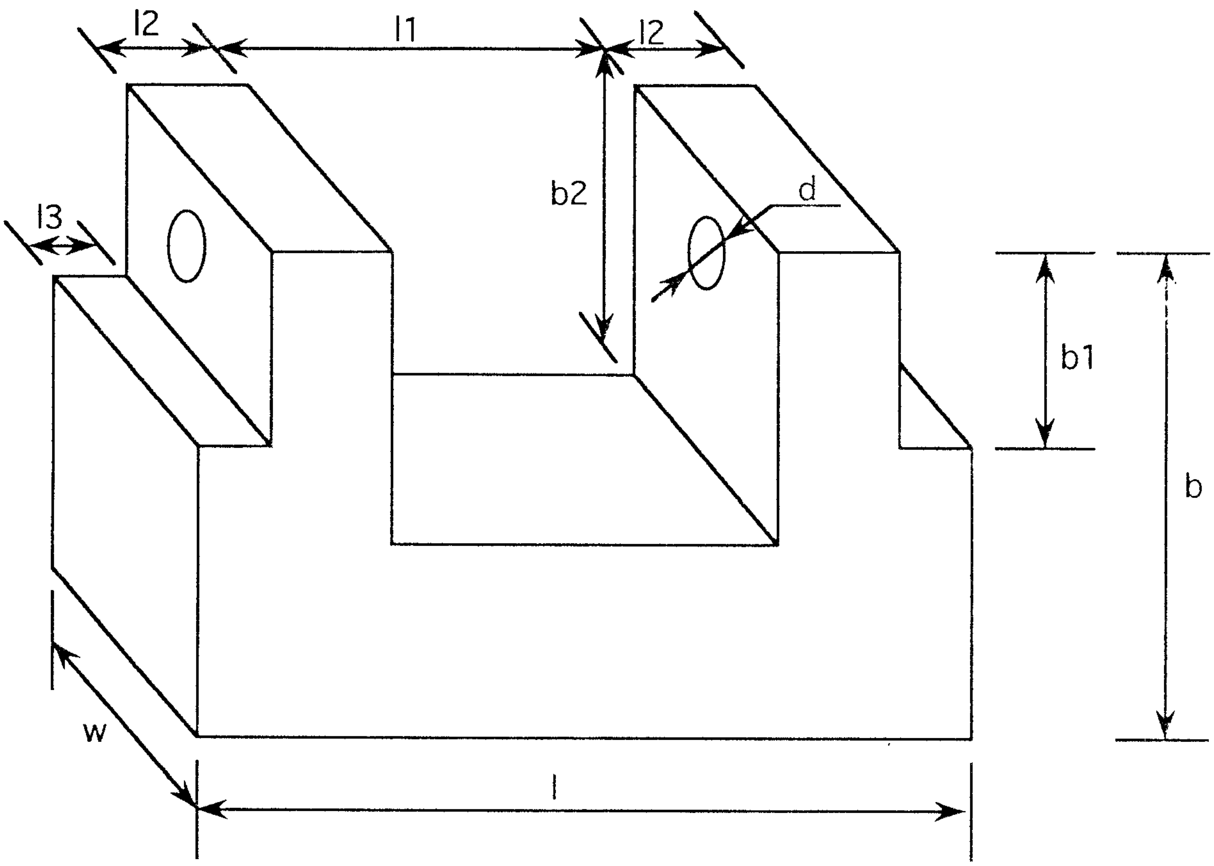
(b) A Trajectory (t)

(c) Planar-Milling-Rectangular

Fig. - 2



(a) Stock (Before Machining)



(b) Finished Part (After Machining)

Fig. 3

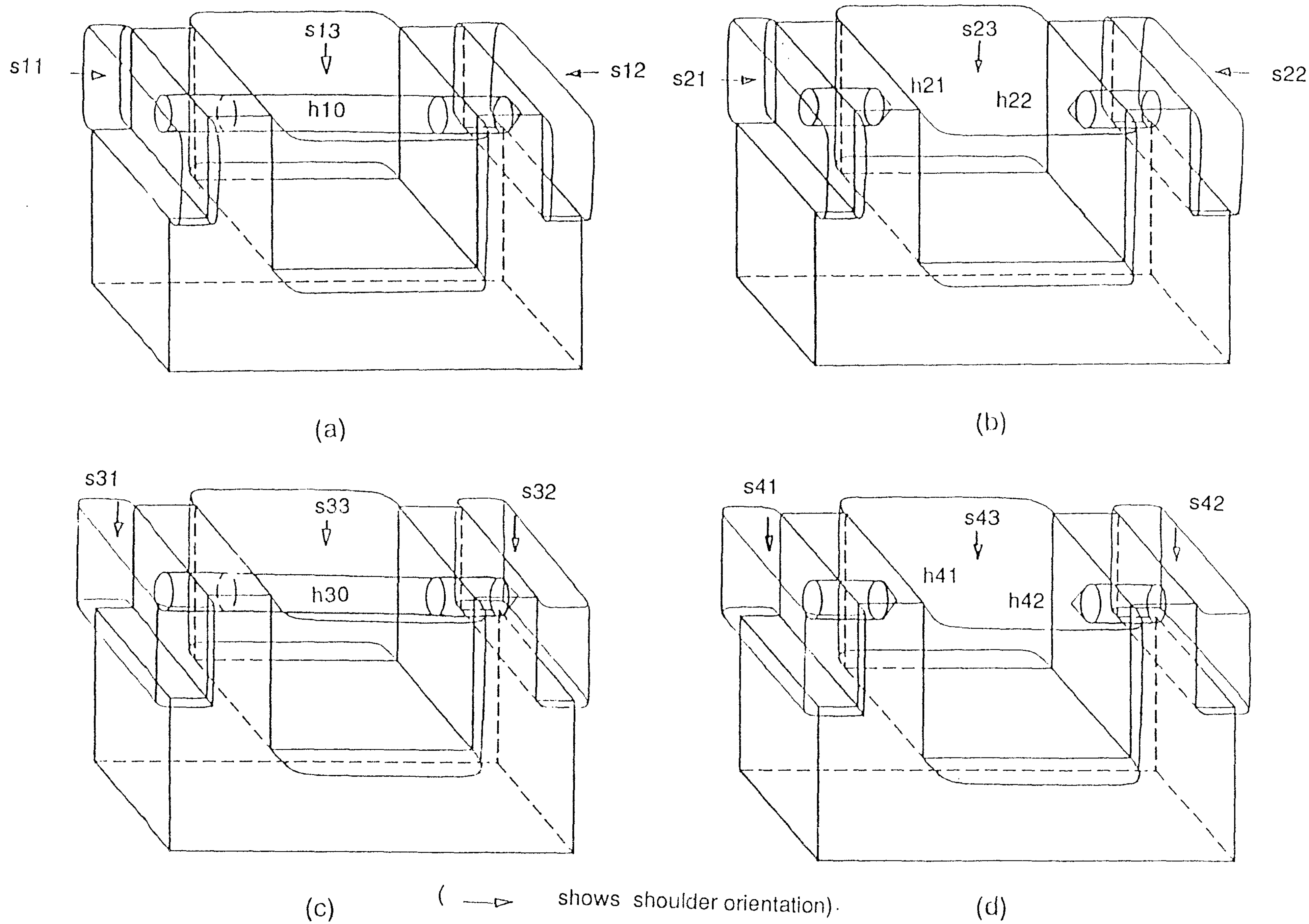
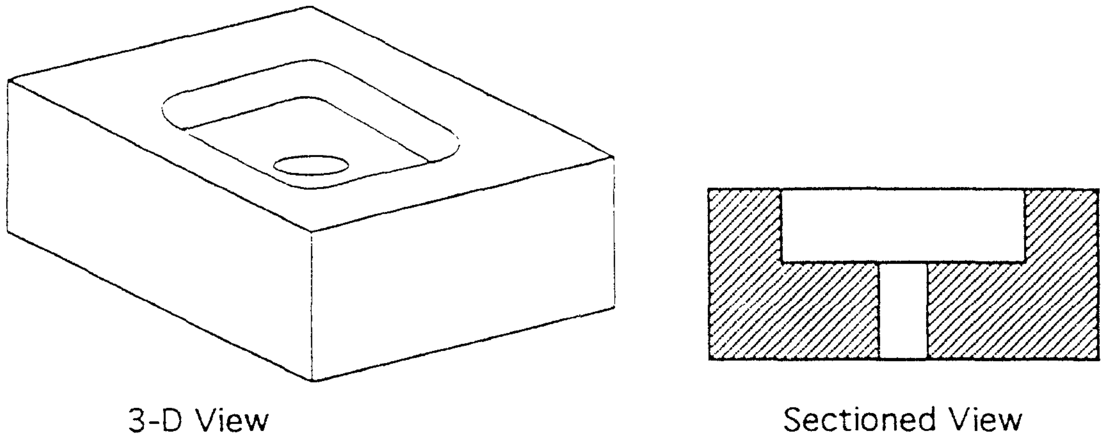
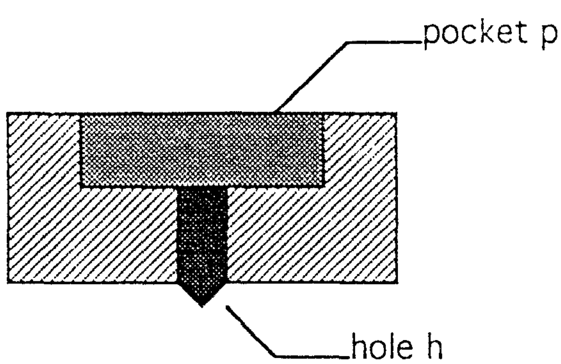


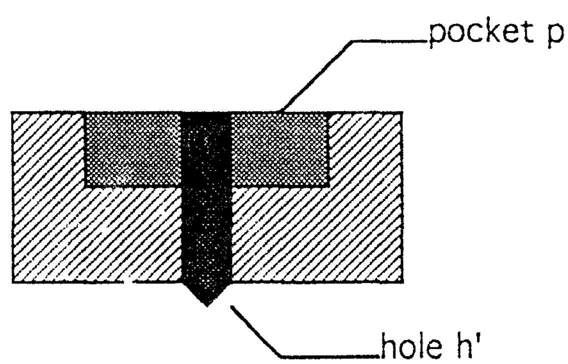
Fig.-4 Bracket Example - Some of the Interpretations



(a) Part



(b) Interpretation 1



(c) interpretation 2

Fig.-5

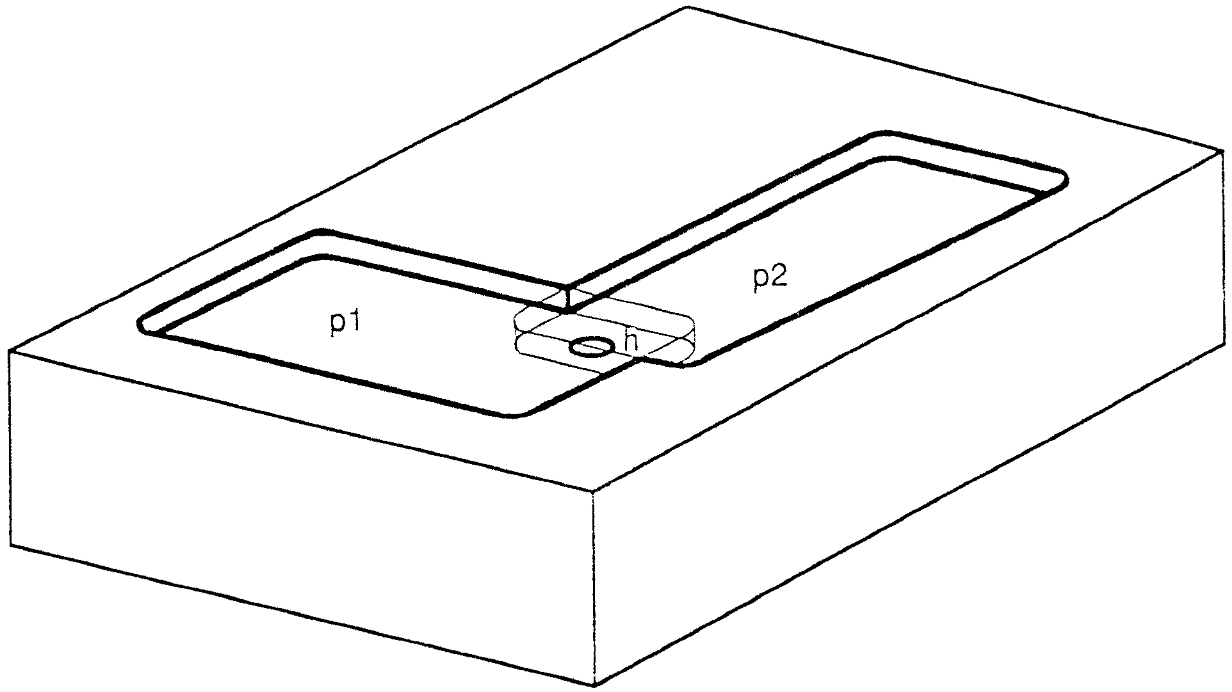
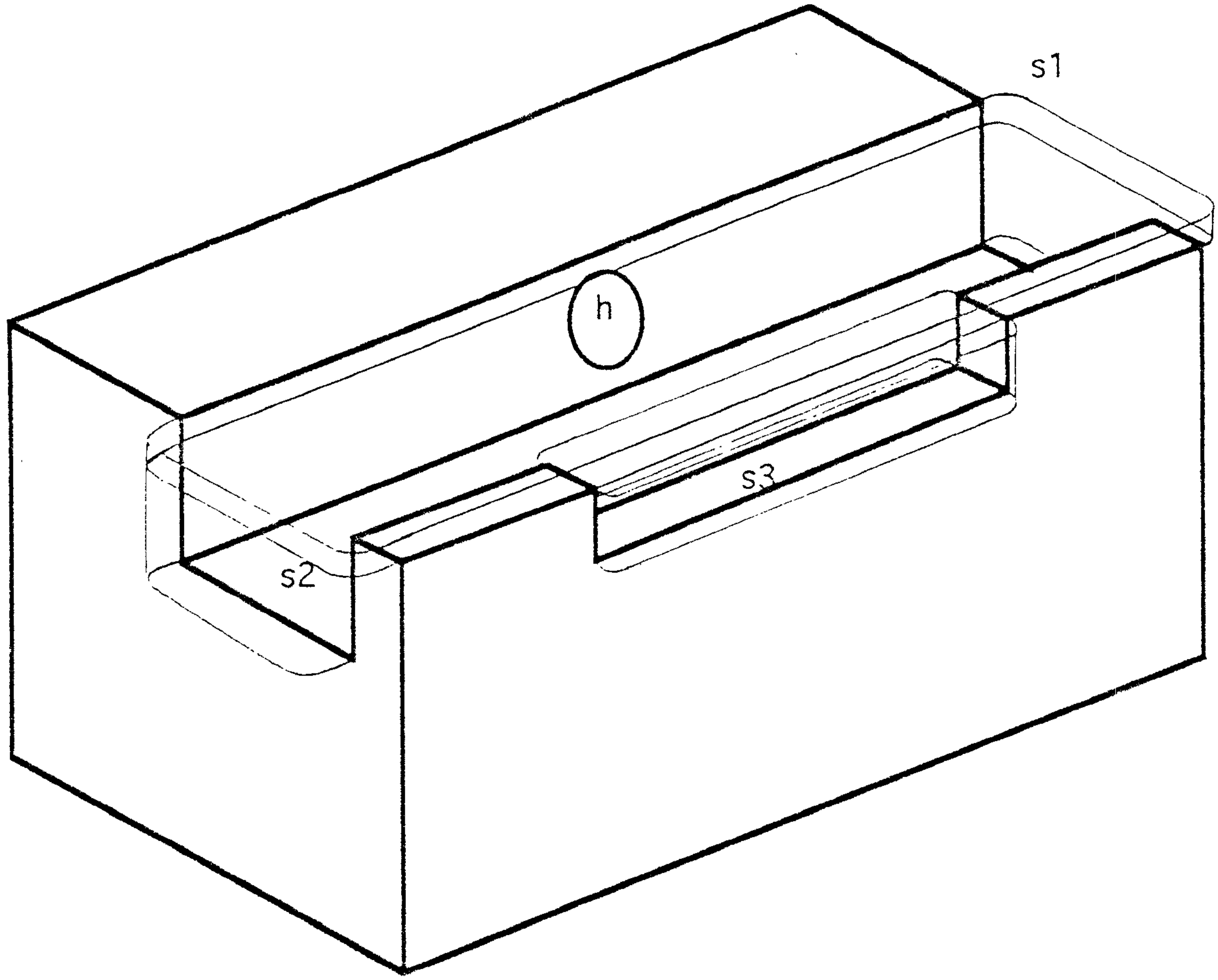
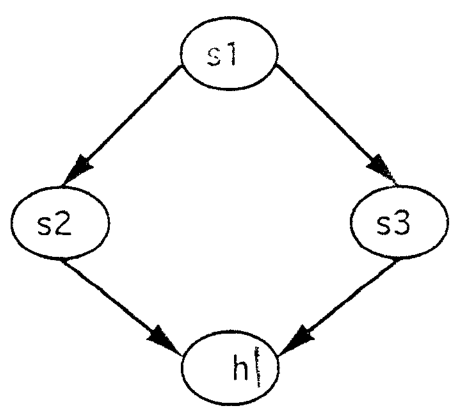


Fig.-6

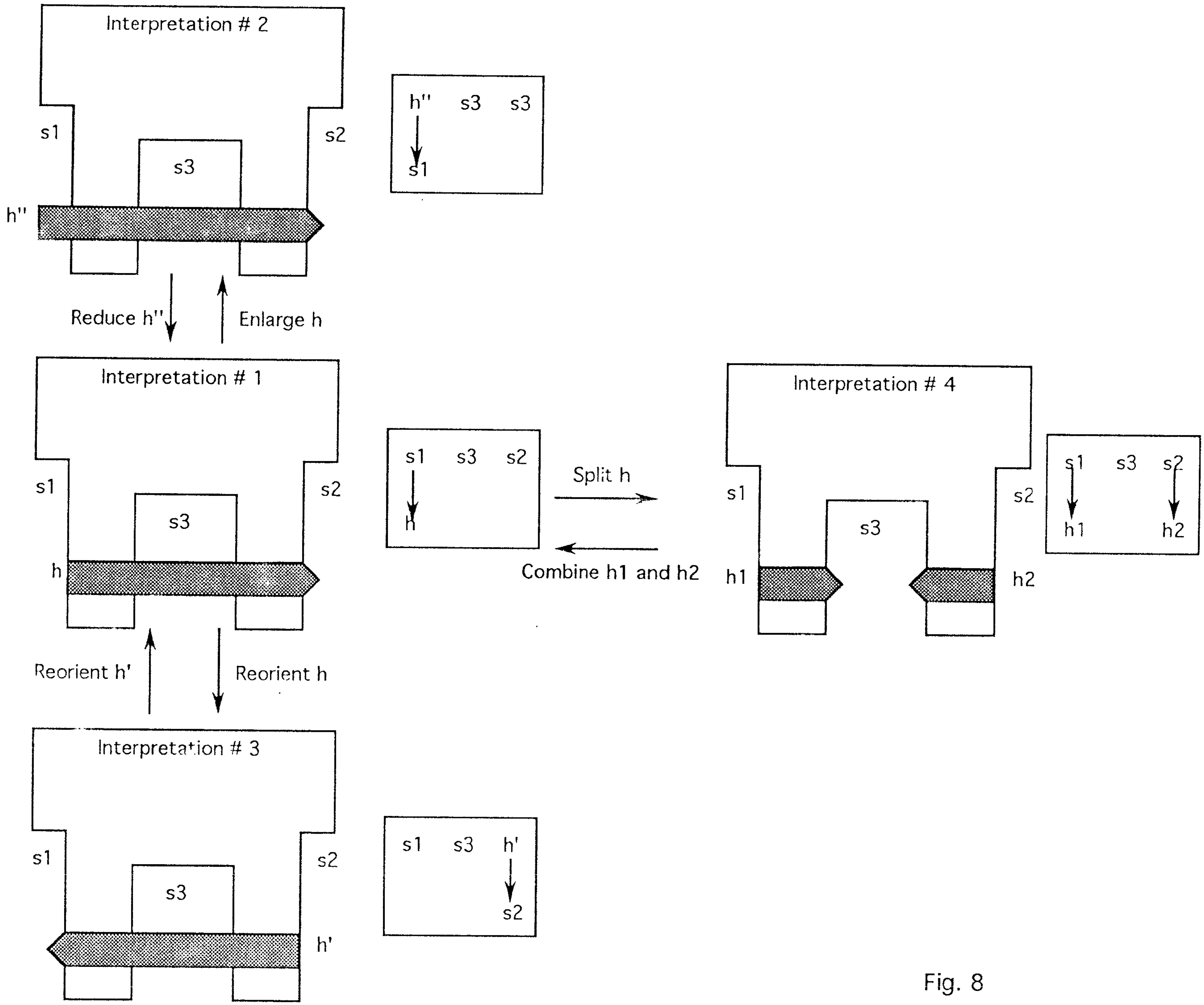


(a) Part



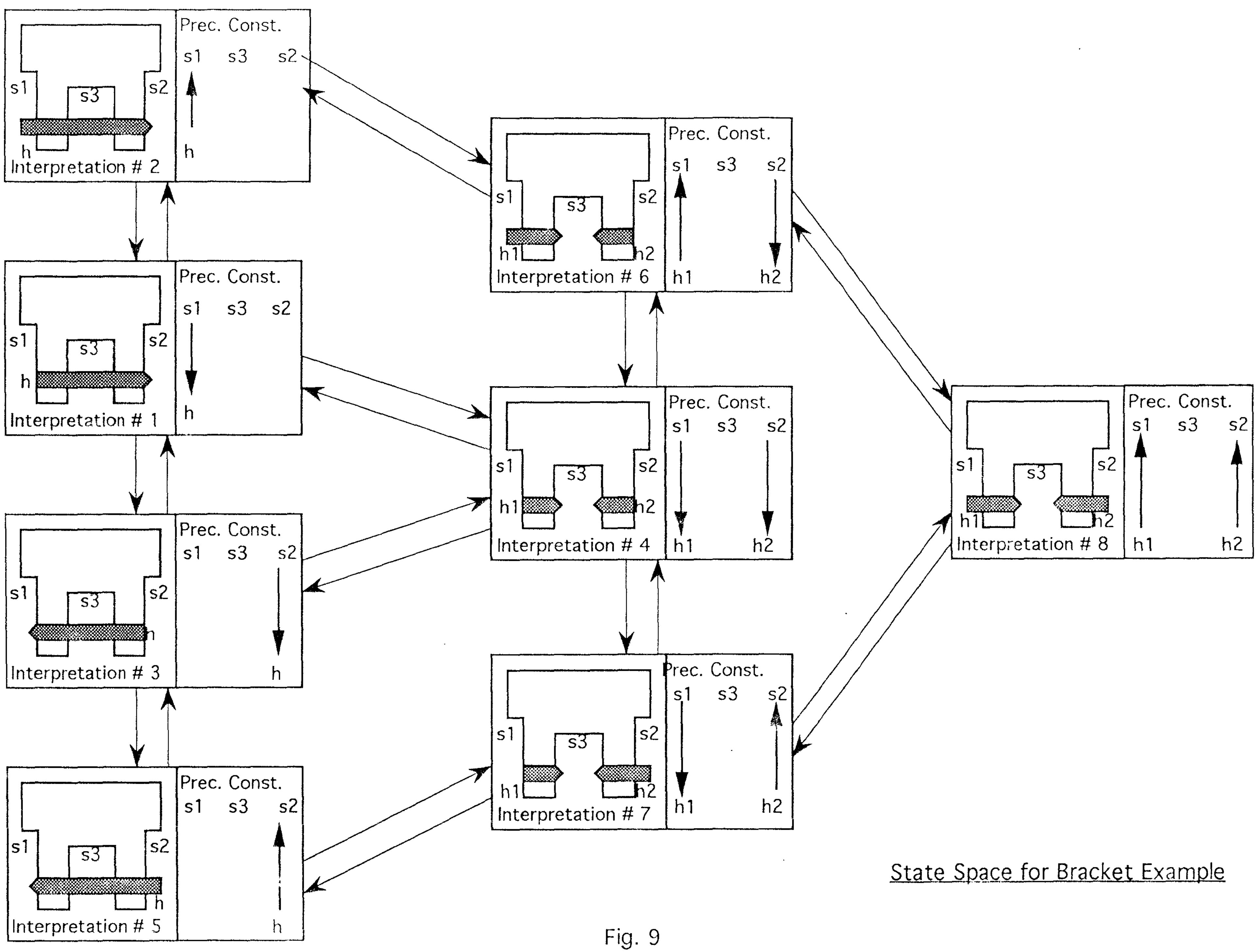
(b) Time-order graph

Fig.-7



Interpretation # 2
s1
s2
s3
h''
h'' s3 s3
s1
Reduce h''
Enlarge h
Interpretation # 1
s1
s2
s3
h
s1 s3 s2
h
Split h
Combine h1 and h2
Interpretation # 4
s1
s2
s3
h1
h2
s1 s3 s2
h1 h2
Reorient h'
Reorient h
Interpretation # 3
s1
s2
s3
h'
s1 s3 h'
s2

Fig. 8



Interpretation # 2
Interpretation # 1
Interpretation # 3
Interpretation # 5
Interpretation # 6
Interpretation # 4
Interpretation # 7
Interpretation # 8
Prec. Const.
s1
s2
s3
h
h1
h2
State Space for Bracket Example

Fig. 9