Building and Structuring a Large Knowledge Base for Computer-Assisted Synthesis Planning

Takashi Nakayama

Department of Information Science, Faculty of Science, Kanagawa University, 2946 Tsuchiya, Hiratsuka-shi, Kanagawa, Japan 259-12

Received May 4, 1995[⊗]

The method of building and structuring a transform knowledge base (TKB) for the computer-assisted synthesis planning system (called SPEK) and the utilization of TKB in the reasoning process are described. The aim of TKB structuring is to enable SPEK to search relevant transforms rapidly from TKB in a target-driven manner based on the result of target analysis. For this purpose, various aspects of TKB are organized in terms of hierarchical, partitioning, or index structures, which realize a flexible and efficient access mechanism to TKB. The reasoning mechanism of SPEK is categorized basically as case-based reasoning, where the relevant transforms are selected from TKB which have a similar reaction site hypothesized on the target structure. The procedure of this similar transform search is also described.

INTRODUCTION

Nowadays, reaction database systems are commonly used in organic chemistry laboratories worldwide. The usefulness of reaction database retrieval systems is generally recognized, though the performance of these retrieval systems and the quality of reaction databases remain as important problems which must be addressed continuously hereafter. Knowledge bases have been said to be critical for transform-based synthesis planning systems such as LHASA and SECS. In fact, consortia of chemical and pharmaceutical companies have been organized to expand the transform knowledge base.[1] However, it is difficult to achieve satisfactory knowledge bases of both sufficient quantity and quality. The situation is the same in other fields of expert systems application, and therefore methodologies for knowledge acquisition and research projects concerning large knowledge base systems have been proposed and developed.[2−5] Such ongoing work indicates that knowledge representation methods and reasoning mechanisms for problem solving remain inapplicable to complex problems. For transform-based synthesis planning systems, the method of representation and utilization of transforms is not completely established, even though they are key to the systems; a great variety of methods have been proposed and tested up to now.[2, 6−8] This paper describes the method of building and structuring a transform knowledge base (called TKB hereafter) which is used by the computer-assisted synthesis planning system based on empirical knowledge (called SPEK hereafter) developed in our laboratory.[9,10]

Transform Knowledge Source. The major part of TKB is built by extracting and arranging transforms automatically from the reaction database (RDB) compiled and built continuously by collaboration of a total of 18 organizations. The reaction center and the atomic correspondence between a starting material and a product are described explicitly in each record of the RDB, while other similar systems which build transform bases from reaction databases include processes which extract the reaction center from each reaction data by recognizing atomic correspondence between a starting material and a product.[7,11] This key information makes it possible for the RDB retrieval system not only to be a simple substructure search system but also to be accessed from the viewpoints of reaction and/or synthesis planning.[12] Furthermore, basic transforms edited manually from textbooks and articles are contained in TKB.[13]

Retrosynthesis Rules. Basic descriptors of a transform are (1) the reaction site substructure, for which matching with a target gives an indication of the transform to be applied, (2) the manipulation for structural transformation of the target, which is performed when the transform is matched with a target, (3) reaction conditions, scope, and limitations used for the evaluation of the validity of the transform application, and (4) types of the transform.[1,9] The features of TKB are that (1) and (2) are unified and represented by the same graph pair and that linguistic expressions of the scope and limitations of (3) are incorporated in the structural representation of the reaction site of (1) as part of graph information through the mechanical extraction of topological relations from the RDB as much as possible. We call this transform element of graph representation the retrosynthesis rule. The set of retrosynthesis rules can be used for both directions of synthetic and retrosynthetic planning, due to the integration of (1) and (2). Incorporation of linguistic expressions of the scope and limitations in graph representation is actually realized in the form of the hierarchical structure of TKB and semantic interpretation of substructures, which contributes to the elaboration of the target-driven search of similar transforms. In the following description of TKB structuring, the retrosynthesis rules and the transforms are treated as identical entities, except when otherwise noted.

Reasoning Method. The reasoning method is a hybrid of rule-based reasoning (RBR) and case-based reasoning (CBR).[14−18] The reasoning cycle in SPEK is as follows: it first analyzes a target and hypothesizes a reaction site, then it searches for transforms in TKB which have a similar reaction substructure (called similar transforms), and if found, it applies the transform to the target, which results in structural transformation into a precursor. The reasoning mechanism of SPEK consists of the definition of similarity, the similar transform search, and the behavior when similar transform search fails. If the genericness of the similar

[⊗] Abstract published in *Advance ACS Abstracts*, September 1, 1995.

0095-2338/95/1635-0885$09.00/0

transform to be sought is high, the transform works as a rule, and its applicability becomes wide, that is, the transform can be found easily in general. On the other hand, if the genericness of the similar transform to be sought is low, the transform is close to the source reaction data (instance), and its applicability becomes narrow, that is, the transform may not be found. In the latter case, the "similar" transform to be sought may often be modified in order to render it applicable to the target. In most cases, the mechanical repetition of this reasoning process results in a combinatorial explosion of the retrosynthesis tree with regard to generated precursors. As a result, a key for applying systems of this kind to practical use is to realize a powerful pruning facility for the TKB search. The search for a similar transform is performed by subgraph matching of the reaction site, that is, the reaction center and its neighboring substructure assumed on a given target. The principal strategy of pruning is to favor the transform which has a larger substructure to be matched. That is, transforms closer to instances are handled prior to ones closer to rules. In addition, the augmented graph-matching mechanism may be used, which is implemented as analogical reasoning through generalization with the use of a semantic dictionary. This analogical matching also virtually expands the matching area.[10]

Structural Similarity of Reaction Site. The similarity between a transform and a target is defined, as mentioned earlier, based on the similarity between the transform's reaction site substructure and the one assumed on the target. Several definitions of similarity concerning chemical graphs have been proposed.[6,19,20] The definition employed in SPEK is based on the idea using maximal common subgraphs. However, the common part between chemical subgraphs does not always give a complete definition of the similarity regarding actual reactivity, because the simple representation of subgraphs does not contain such information as that concerning equivalent atoms and atomic groups and the effects of surrounding substructures. Hence, in SPEK, the concept of an augmented subgraph, called a feature graph, is defined as a primary criterion of the similarity of reaction sites, which represents the reaction center and the features of its surrounding area. Furthermore, subgraph matching through generalization using the semantic dictionary also contributes to a more general definition of similarity.[10] Information concerning the presence or absence of, e.g., functional groups and ring systems, corresponding to keywords in texts, also expresses some kind of similarity.[20] The feature graph is composed of graph representation of reaction sites together with information on the presence or absence of particular substructures, path information, and topological relations.

Structuring Methods. The objective of TKB structuring is to enable SPEK to select appropriate transforms quickly based on the result of target analysis. The propriety of transform selection is evaluated based on the similarity of reaction sites mentioned above. TKB is structured by the following three approaches: (1) hierarchical structuring and partitioning of the main file, (2) use of various index files as access structures, and (3) virtual multiplication of transforms by semantic interpretation of substructures.

Automatic target-driven TKB access is realized through these structures. TKB hierarchy consists of five layers: primitive, core, feature, refined, and maximal TKB layers. The first access in SPEK is directed at the feature TKB. The accesses to other TKBs are set dynamically according to the first access, which leads to the navigation of the entire TKB network. Virtual multiplication of transforms is realized in each TKB layer.

Table 1. Atom Classes of Reaction Data in RDB

class	starting material	product
1 (ACR1)	connection relation unchanged	connection relation unchanged
2 (ACR2)	bond multiplicity changed	bond multiplicity changed
3 (ACR3)	adjacent atoms changed	adjacent atoms changed
4(ACR4)	leaving atoms	added atoms

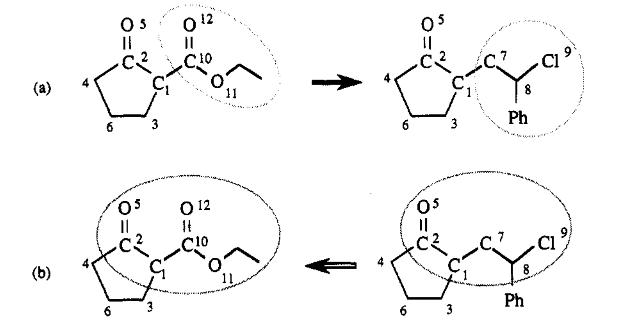


Figure 1. (a) Atom classes from the viewpoint of reaction. C1 is ACR3 (i.e., class 3 both in the starting material and the product) and is a reaction center atom. Atoms enclosed by dotted lines are ACR4, that is, atoms C10, C11, and C12 leave the starting material, while atoms C7, C8, and C9 of class 4 appear in the product. All other atoms are ACR1 which do not participate in the reaction in terms of structural change. (b) Atom classes from the viewpoint of transform. C1 and C7 in the product, and C1 and C10 in the starting material are tca. C2=O5 and Cl in the product, and C2=O5 and ethoxycarbonyl (C10, C11, C12,...) in the starting material are aaa. An atom C8 on the path C7−C8−Cl in the product is opa.

REPRESENTATION OF REACTION DATA AND TRANSFORMS

Reaction Data. The reaction database (RDB) is currently organized from about 50 000 specific reaction data, each of which consists of a reaction scheme (structural data) and text data. The structural data are described from the viewpoint of constituent atoms, where the atomic correspondence between a starting material and a product has been previously established.[8] The constituent atoms of starting materials and products are categorized into four classes, as shown in Table 1, based on their relation to the reaction. Each of the four classes is called an ACR (atom class in reaction) and is discriminated between starting materials and products. The atoms which do not change their adjacency (connection relations with adjacent atoms) between starting materials and products belong to ACR1 (class 1). ACR2 contains atoms whose adjacency changes between starting materials and products, that is, adjacent atoms are substituted or removed. ACR3 contains atoms having a bond attribute, called multiplicity in SPEK, which changes between starting materials and products. Multiplicity is simply the number of double bonds attached to an atom, where a triple bond is counted as multiplicity 2 and an aromatic bond as 0.5. Atoms belonging to ACR2 and ACR3 are called reaction center atoms. ACR4 contains atoms which are removed from starting materials and are added to products. ACR4 may be referred to, if necessary, in order to obtain leaving groups when SPEK generates precursors in the reasoning process. Primitive transforms, described in a later section, are immediately obtained from RDB only if these classes are extracted. An example of reaction data is shown in Figure 1(a).

Table 2. Items of a Transform Record[b]

items	meaning
registry number	identifier of a transform in a TKB layer
name	transform name (optional)
supertransform[a]	a transform from which this transform is derived
subtransform[a]	a group of transforms in which this transform is commonly possessed
reaction instance	a group of reaction instances from which this transform is extracted
type	type of transform
pattern	pattern of transform -
LHS	left-hand side of reaction site (EMN)
RHS	right-hand side of reaction site (EMN)
favorability	certainty factor of transform
scope and limitations	scope and limitations
conditions	reaction conditions
editor	expert's name who edited this transform

[a] Variations of super- and subtransforms in each TKB: primitive TKB: supertransform = none, subtransforms = pointer to core TKB; core TKB: supertransform = pointer to primitive TKB, subtransforms = pointer to feature TKB, pointer to maximal TKB; feature TKB: supertransform = pointer to core TKB, subtransforms = none; maximal TKB: supertransform = pointer to core TKB, subtransforms = none. [b] Categories of TKB used above are explained in section Method of Structuring.

Transforms. A transform record consists of items shown in Table 2. The representation of records is almost the same as the so-called frame representation. TKB is constructed hierarchically based on the inclusion relation of reaction sites. The items supertransform, subtransform, and reaction instance in Table 2 represent index sets pointing to higher layers, lower layers, and RDB, respectively. Figure 2(a) shows an image of TKB hierarchy, where the terms "primitive", "core", "feature", "refined", and "maximal" indicate layers of TKB hierarchy, of which members are called primitive transforms, core transforms, feature transforms, refined transforms, and maximal transforms, respectively. For instance, if some primitive transform is included as a common substructure of several core transforms, these core transforms are set as subtransforms of the primitive transform. These super- and subrelations hold between the layers linked by solid arrows shown in Figure 2(a); there are some variations in the actual record format among layers, as indicated below Table 2. Record items LHS and RHS represent the structures of reaction sites which may be called transform sites. LHS represents the reaction sites of starting materials in the reaction, and RHS those of products. Thus, the structural transformation scheme for transforms is expressed as RHS → LHS in each hierarchical layer. However, actual transformation rules are common to those of the primitive layer, while LHS and RHS in other layers contain surrounding substructures only for premises of rule application. The general framework of structural transformation rules is shown in Figure 2(b). An example is shown and explained in the next section and in Figure 4. This paper concerns the mechanism of selecting a transform whose reaction (transform) site is as close as possible to the proper one, and hence the term transform used hereafter means LHS or RHS. The constituent atoms of a transform are categorized into three classes called ACT (atom class in transform), although LHS and RHS vary depending on the layer to which they belong. ACTs are (i) tca (transform center atoms), member atoms of the reaction center and their adjacent ACR4 atoms; (ii) aaa (affecting atoms apart from reaction center), member atoms of functional groups apart from the reaction center, where there are no other functional groups on or adjacent to the shortest path between the reaction center and the functional group in question, provided that the path length (the number of constituent bonds of the path) is within three; and (iii) opa (on-path atoms between tca and aaa), the path is the one used for defining aaa. An example of ACT categorization is shown in Figure 1(b). Atoms other than those in these three classes may be included in transforms of some layers.

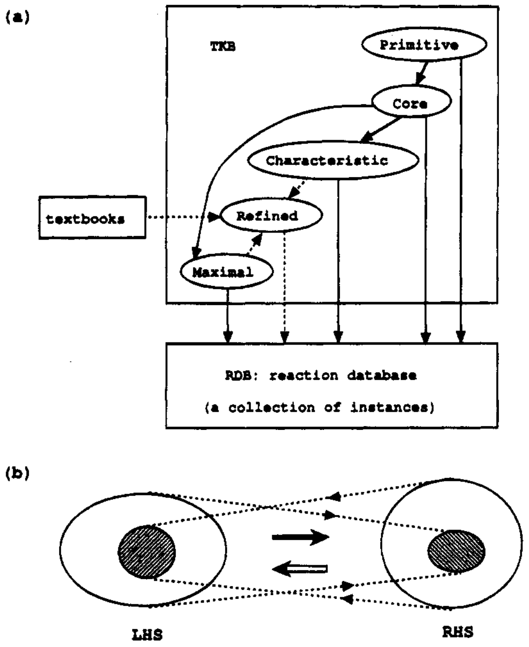


Figure 2. (a) Hierarchy of TKB and RDB. The entire TKB is organized hierarchically based on the inclusion relation with regard to reaction sites, where RDB is a base as well as a source of TKB. The strict inclusion relation holds only for primitive, core, and maximal TKB layers. Feature and refined TKBs are approximately located at the levels indicated in the figure, based on their source. The dotted arrows to refined TKB suggest its source data, and arrows to RDB suggest that RDB is accessed from all the TKB layers. (b) General framework of structural transformation rules. The actual parts in which the connection relation changes are indicated by hatched areas, which correspond to primitive transforms. The parts surrounding the hatched areas correspond to areas which are expanded in lower layers. Each transform contains LHS and RHS with the surrounding area.

METHOD OF STRUCTURING

Structure of Main File. Figure 3 shows a framework of the hierarchical structure of the entire TKB and partitioning structure of each TKB layer, which forms a grid file.

A. Hierarchical Structure. Hierarchy in TKB reflects the inclusion relation between subgraphs of reaction sites (i.e., reaction center atoms and their surrounding substructures) which are extracted from reaction data by several different methods. The inclusion relation is given by the extent of the surrounding substructures assumed in those extracting methods. In general, however, only partial order holds among subgraphs, and hence an ad hoc procedure to expand surrounding areas is necessary to maintain total order. Since there are no definitive algorithms to determine correct reaction sites automatically, the surrounding areas for involvement in the reaction are extracted in a manifold manner from one reaction instance. The following five types of transform extraction are provided in SPEK, which forms the hierarchical structure of five TKB layers.

(1) Primitive TKB. This is generated immediately from RDB, by extracting tca (ACR2, ACR3, and ACR4 attached

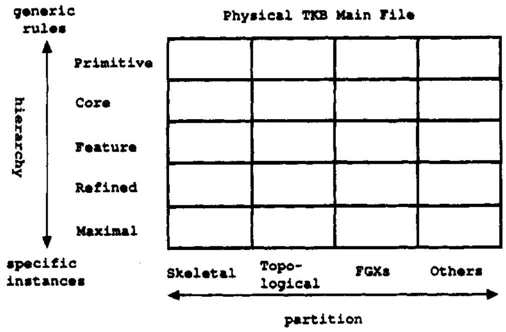


Figure 3. Physical files for TKB are made separately for each hierarchical layer: primitive, core, feature, refined, and maximal TKBs. SPEK can navigate these layers through super- and/or subtransform relations. Each layer is partitioned into four categories: skeletal, topological, FGXs, and others. As a result, the entire TKB forms a grid file, where every combination of layer and category gives a unit of search space.

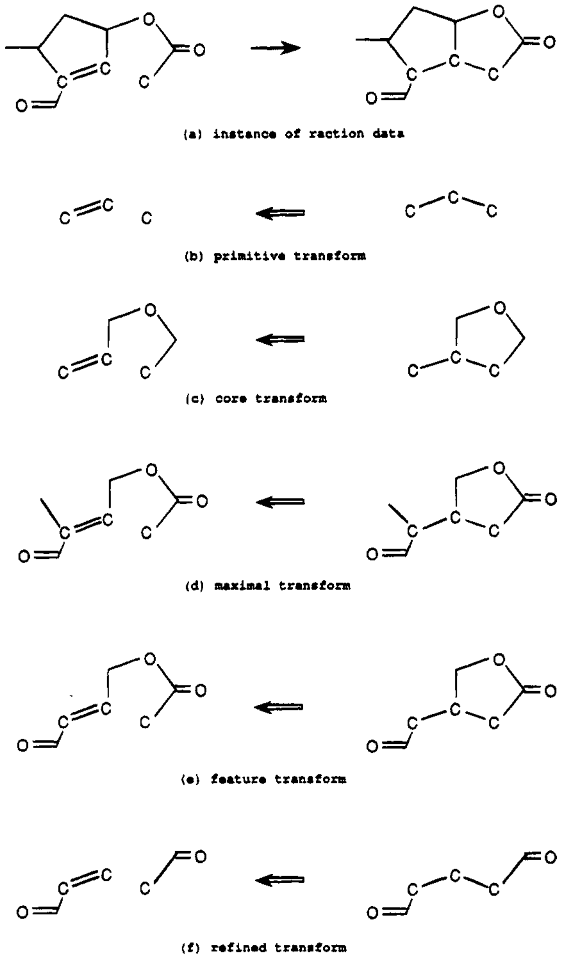


Figure 4. Examples of transform site extraction: (a) reaction data, (b) primitive transforms, (c) core transform, (d) maximal transform, (e) feature transform, and (f) refined transform.

to ACR2/ACR3). Figure 4(b) shows an example of a primitive transform extracted from the reaction instance shown in Figure 4(a). All the transforms in other layers contain a primitive transform which represents the common structural transformation.

(2) **Core TKB.** Roughly speaking, core TKB is a collection of transforms produced by modifying primitive TKB so as to make disconnected transforms (disconnected graphs) connected. Some transforms in primitive TKB contain disconnected graphs in their reaction sites. Disconnected graphs can be either of LHS (starting material part) or RHS (product part) or both. When only one side of LHS or RHS is disconnected, it is permissible to leave it disconnected unless the side is matched with targets. However, it is necessary to dissolve the disconnection in order to apply transforms in both retrosynthetic and synthetic directions. In fact, when disconnected graphs are extracted, it is not clear whether they are subgraphs of originally different components or subgraphs of the same component, because there are some reaction data which comprise multiple starting materials/products. Thus, for disconnected graphs, the shortest path formed by only ACR1 atoms is sought from among the component subgraphs, although it is not always unique. A connected subgraph is then made of those subgraphs and the path, giving a new reaction site. The transforms generated from primitive ones in this manner are called core transforms. When only one side of LHS or RHS is disconnected, the corresponding atoms on the path are added to the other side. There are cases where solvents, catalysts, and reagents are described as components, which are excluded in the extraction process. However, multiple transforms are generated in the case of reagents which cannot be excluded automatically. An example of core transforms is shown in Figure 4(c).

(3) **Maximal TKB.** This is generated from primitive transforms by extending the reaction site (i.e., core site) using Wilcox and Levinson's algorithm.[21] Since the algorithm extends the reaction site area continuously, functional groups far from the original reaction site may be omitted. On the other hand, atoms which have no influence on the reaction may be incorporated. Nevertheless, this type of transform is included as a TKB layer in the current version of SPEK, because of its simple process. Figure 4(d) shows an example of a maximal transform extraction.

(4) **Feature TKB.** While primitive, core, and maximal transforms consist of plain chemical graphs which contain reaction centers, feature graphs are defined as chemical subgraphs which have attributes other than ordinary attributes inherited from reaction data, such as properties of neighboring areas and topological relations to surrounding substructures.

(a) Atoms of a feature graph include the following: (i) tca, transform center atoms mentioned in the previous section, i.e., reaction center atoms (ACR2 and ACR3 atoms) and their adjacent ACR4 atoms; (ii) ACR1 atoms which are on the shortest path between disconnected components if the reaction center is disconnected; and (iii) aaa and opa.

(b) The skeleton of a feature graph is a subgraph made of the constituent atoms described above and bonds between those atoms. Besides the attributes inherited from their parent graph, the following five attributes are also added to the constituent atoms extracted as above: (i) degree in the parent graph, (ii) atom class in the transform (ACT), (iii) membership to smallest rings (the sizes of smallest rings to which the atom belongs), (iv) membership to functional groups (the identifier to which the atom belongs), and (v) ring multiplicity.

In Figure 1(b), substructures enclosed by dotted circles correspond to feature graphs of the starting material and the product. An example of feature transform extraction is shown in Figure 4(e).

(5) **Refined TKB.** This is generated through manual encoding from textbooks and/or articles or editing transforms compiled from RDB as described above. This may be situated between the core and maximal TKB, near the feature TKB in the hierarchy. In actuality, however, it is indepen-

dent of the hierarchy, because the source of transforms is different from that of the other four layers, or the generation process does not take into account the inclusion relation concerning reaction sites when editing core/primitive transforms. It is organized in the main file as a layer in the hierarchical structure. Figure 4(f) shows an example of refined transform.

Each transform of TKB layers is accessed through index files concerning reaction site structures (extended Morgan name). Each transform is linked to corresponding super- and/or subtransforms as well as to reaction instances. Figure 5 shows a conceptual diagram of access from a target to TKB layers. In reality, TKB hierarchy exists in both LHS and RHS; here only the RHS side is implemented.

B. Partitioning Structure. Each layer of the hierarchy is partitioned into classes based on the character of structural transformation performed by transforms in that layer. When sequential search is carried out in a TKB layer, this partitioning reduces the search space and thus enables users to specify one of these partitioned classes as a strategy in an interactive session of this system. There are four partitioning classes.

(1) **Skeletal Transforms.** This class consists of transforms which result in major changes in skeletal structure, such as multiple bond disconnection and/or reconnection. In other words, transforms in this class do not have common reactivity or a common structural feature; rather they are a collection of important transforms which should be applied preferentially if there is some degree of similarity with the target. This class is formed by selecting typical and powerful transforms manually, which may be categorized into topological transforms described below from the viewpoint of the character of topological change.

(2) **Topological Transforms.** This class includes transforms which disconnect a skeletal bond, that is, C−C bond or Q−C bond (Q is a heteroatom). The disconnections are recognized automatically as a structural transformation pattern when transforms are extracted from RDB.

(3) **Functional Group Transforms.** This class includes transforms which do not involve skeletal changes but perform functional group interchange, addition, removal, and modification, called FGI, FGA, FGR, and FGM, respectively, which are denoted generically as FGX. These are also recognized automatically when they are extracted from RDB with the use of a functional group dictionary.

(4) **Others.** Transforms which do not fall under any of the three classes given above are grouped as "others". The members of this class are not fixed but may be relocated if a better class is found or designated as a new class. For instance, stereochemical transforms are discriminated by attributes at present, but it may be better to group them into one class. The horizontal direction of Figure 3 shows the partitioning of TKB layers.

Structures as Access Methods. The following classifications of transforms are implemented as index files and are used for access methods to both TKB and RDB.

A. Classifications Based on Transform Type. Transforms are classified based on the types of structural transformation which they produce. This corresponds to formal classification of reactions based on the structural changes between starting materials and products. However, the names of the types of structural changes represent the meaning from the viewpoint of transforms, that is, the changes from the product side to starting material side. The types are recognized automatically when transforms are extracted from RDB.

(1) **Interchange.** This corresponds to substitution in reactions. Typical examples are functional group interchange (FGI) and functional group addition (FGA). Skeletal structures are unchanged after this type of transform, except for the interchanged atoms. The transformation as a graph is recognized such a change as both the removal of an attached atom or a substructure and the addition of another atom or a substructure occur at the same node.

(2) **Removal.** This corresponds to addition in reactions. Skeletal structures are unchanged after this type of transform, except for removal of atoms/subgraphs and the bond order of counterpart atoms. The transformation as a graph is recognized as disconnection of an edge with multiplicity change (increase).

(3) **Addition.** This corresponds to elimination in reactions. Skeletal structures are unchanged after this type of transform, except for the added atoms/subgraphs and the multiplicities of reaction center atoms. The transformation as a graph is recognized as edge generation with multiplicity change (decrease).

(4) **Rearrangement.** This corresponds to rearrangement in reactions. In general, skeletal structures change after this type of transform. The transformation as a graph is recognized as the disconnection and the generation of edges on the same graph.

(5) **Disconnection.** This corresponds to condensation or cyclization in reactions. The transformation as a graph is recognized as skeletal changes resulting from edge disconnection, which is different from interchange. Since the disconnection of rings can be detected automatically, it is possible to subdivide this type into subclasses corresponding to condensation and cyclization. In condensation, two graph components are generated by edge disconnection, whereas the degree of connection of the graph decreases in cyclization. Furthermore, it must be considered that the number of disconnected edges may not always be only one.

(6) **Reconnection.** This corresponds to ring opening, shrinking, or enlargement in reactions. The transformation as a graph is recognized as a skeletal change which gives rise to new rings.

(7) **Others.** Transforms which do not belong to any of the classes above, or which cannot be classified by the currently implemented algorithm in SPEK, are grouped into this class.

B. Classifications Based on Structural Fragments. Reaction centers, ring systems, and functional groups are extracted from reaction instances (RDB) as existential information, which give access paths from substructure-based index files. These index files are ordinarily used for access to RDB, while TKB is accessed primarily through a feature graph index file which should reflect the entire reaction site, as described later.

C. Classifications Based on Topological Features. Although structural features such as ring systems and functional groups are, at most, existential information, they are commonly used in many chemical structure database systems. In addition to such information, the following topological features, which represent the positional relationship among ring systems and/or functional groups in a whole structure, are provided as index files in SPEK. They are also used mainly for access to RDB.

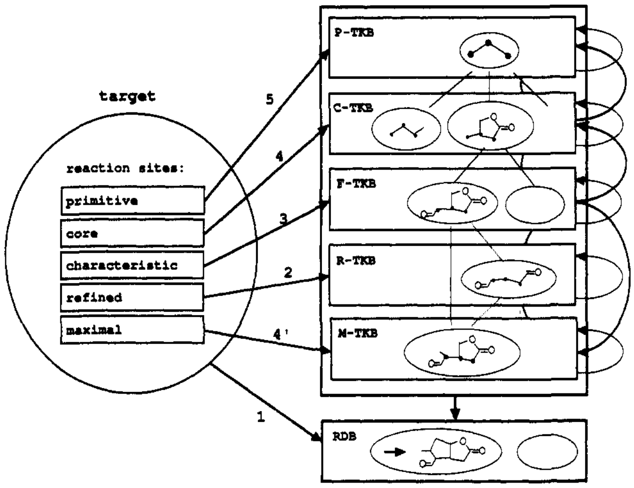
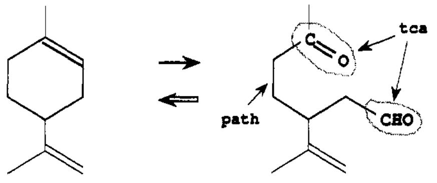


Figure 5. Access from a target to TKB layers is indicated by arrows. Numbers attached to the arrows suggest the access order. Arrows from one layer to another show the manner of navigation in TKB layers. Before navigating to other layers, the current layer may be searched again by generalizing the reaction sites, which are indicated by arrows directed to themselves. Core/primitive TKBs are accessed when search fails in the other layers. Navigation to maximal TKB is carried out when there are multiple matched transforms in the previous TKB.



structural fragments = {>C=O, -CHO, -C=C-}

topological relations = {path(>C=0, -CHO, 5, -),
path(>C=O, -C=C-, 4, -),
path(-CHO, -C=C-, 3, -)}

Figure 6. An example of structural fragments and topological relations.

(1) Distance between Functional Groups. The distance between two functional groups is defined as the number of edges contained in the shortest path between the two.

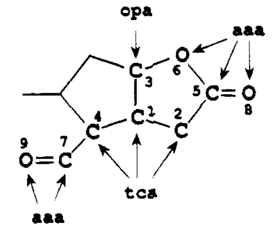
(2) Position Isomerism in Benzene Rings. The positional relationship between two locants corresponding to ortho- (o), meta- (m), and para- (p) is expressed as the distance along the path on a benzene ring. The notations o, m, and p are used in place of distance values 1, 2, and 3 in order to indicate that the path is in a benzene ring.

(3) Geometrical Isomerism Derived from Double Bonds/Simple Rings. Since this is not described explicitly in RDB, the cis-/trans-relationship is not extracted automatically at present. However, the slot for this item is prepared so that the information can be set up by the user.

In short, the topological relationships described above are used to set up conditions to the path between functional groups, and are expressed formally as follows:

path (f1, f2, distance, cis/trans)
f1, f2 : functional group identifier
distance : 1(=o), 2(=m), 3(=p), 4, 5, ...
cis/trans: cd...cis- for double bond
td ... trans- for double bond
cr ... cis- for ring
tr ... trans- for ring

Figure 6 shows an example of the extraction of structural fragments and topological relationships.



	1	2	3	4	5	6	7	8	9	
from	-	1	1	1	2	3	4	5	7	
r.c.										(5,6)
bond	-	1	1	1	1	1	1	2	2	1
atom	C	C	C	C	C	O	C	O	O	
mult	0	0	0	0	1	0	1	1	1	
acr	3	3	1	2	1	1	1	1	1	
degp	3	2	2	2	3	2	2	1	1	
act	c	c	p	c	a	a	a	a	a	
memr	5,5	5	5,5	5	5	5	-	-	-	
memf	-	-	-	-	78	-	78	78	78	
r.m.	2	1	2	1	1	1	0	0	0	

Figure 7. Examples of extended Morgan name (EMN) for a feature graph. A feature graph consists of atoms numbered 1, ...,9. The attributes below "degp" are characteristic offeature graphs: degp, degree of an atom in its parent graph; act, atom class in transforms (c: tca, a: aaa, p: opa); memr, membership to smallest rings; memf, membership to functional groups; and rm, ring multiplicity. For instance, atom C1 is a member of two rings (five-membered rings, in this case) of identifier 5, and atom O8 is a member of a functional group of identifier 78.

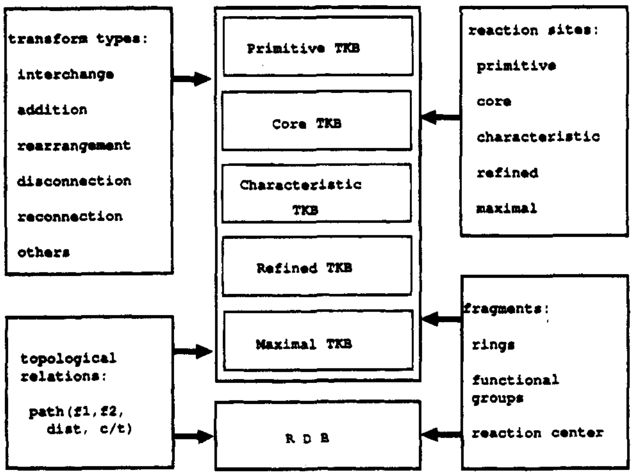


Figure 8. Structure from the viewpoint of index files. Relationship between TKB and index files is illustrated. Access from reaction sites shown at the upper right of the figure is detailed in Figure 5.

D. Classifications Based on Reaction Sites. TKB is constructed hierarchically as described before, and each layer of the hierarchy is accessed by the extended Morgan name (EMN) of the reaction sites. That is, EMN stands for the key of each transform. Thus, this file constitutes a primary index, whereas the three files described above give secondary indices. However, the representation of EMN for the feature TKB is somewhat different from that for other TKB layers. The key to the feature TKB is given in the form of feature graphs, that is, the representation of feature graphs is an extended form of an ordinary EMN. These keys can be viewed as an access method for similar instance search. An example of EMN of a feature graph is shown in Figure 7. The EMN format of other layers is given by the one from which five attributes under "acr" are excluded. EMN with all attributes, that is, that of a feature graph, is called augmented EMN when necessary. The entire constitution of TKB structured in terms of index files is shown in Figure 8.

Virtually Multiplied Structure by Semantic Interpretation of Substructures. Generic representation of the reaction site allows flexibility in matching with targets. The reaction sites obtained through the mechanical extraction process described earlier are specific substructures, since their source instances are specific. Currently, experts carry out matching with targets through generalization of substructures using their knowledge on the reactivity of substructures, since the general properties of atoms and substructures are known and systematically organized. The generalization function of substructures is implemented in SPEK by making use of a semantic dictionary.[10] The semantic dictionary may be seen as a kind of thesaurus or ontology whose items are comprised of chemical structures and/or chemical properties, organized independently of TKB, to which new items can be added if necessary. Surrounding substructures of reaction sites may be described generically using terms defined in the semantic dictionary, although the description itself must be carried out by experts.[9] On the other hand, specific structures acquired automatically from RDB can be changed to generic structures virtually by navigating the generalization hierarchy of the semantic dictionary, if terms are provided for those substructures. This semantic interpretation function is common to all TKB layers. The interpretation mechanism of substructures using the semantic dictionary is reported in ref 10.

REASONING MECHANISM

As mentioned earlier, the reasoning performed in SPEK is a combination of RBR (rule-based reasoning) and CBR (case-based reasoning), but the main reasoning mechanism is CBR, since TKB (transform knowledge base) itself is constructed hierarchically in which RDB, i.e., a set of reaction instances, is placed in the base layer.

CBR in SPEK. The outline of the general process of CBR is as follows.

(1) Analyze and extract features of a given problem. A rule set is provided for feature analysis. Enumerate problems to be expected.

(2) Retrieve the instance which best matches the given problem.

(3) Form a solution from the retrieved instance by modifying the parts which differ from the problem.

(4) If the retrieved instance is not modified well, another instance is retrieved or the rule set is modified by domain knowledge so that the instance is manipulated to be a solution. There are cases in which no solutions can be found even after these procedures.

(5) Store the retrieved instance as a successful case into the case base.

Compared with the process above, the reasoning process of SPEK is as follows.

(1) Analyze a given target. A set of procedures for extracting structural features which are described in the previous section is prepared.

(2) Search for a transform, called a similar transform, whose LHS or RHS resembles the reaction site assumed on the target.

(3) Apply the transform to the target to produce a precursor. Modify the transform to make it applicable, if necessary.

(4) If the produced precursor is available, the reasoning process ends. Otherwise, set the precursor as the next target and return to (1).

It is apparent that this process corresponds to the general CBR process above. In SPEK, the problem is given in the form of a target to be synthesized. Structural analysis of a target as a chemical graph is nothing less than feature analysis of a problem. Accompanying problems expected are the detection of correct differences between, for example, a target and a transform or functional groups to be protected. TKB and RDB correspond to a case base, as do instances to transforms. Instances in RDB (i.e., reaction data) are also referred to indirectly when information such as yield and reaction conditions is requested. Modifications of reaction sites assumed on targets through FGXs and generalization of substructures correspond to those of retrieved instances. However, solutions are not stored in TKB. Characteristics of SPEK are feature analysis of targets and subsequent access methods to the case base (TKB). As noted earlier, SPEK has a hybrid reasoning function of RBR and CBR, where it becomes RBR if a transform used is highly abstracted (the transform is close to a rule) and becomes CBR otherwise (the transform is close to an instance). Actually, TKB is not separated distinctly into the rule base and case base, but the degree of abstraction of transforms is gradual, corresponding to TKB hierarchy, which allows flexible reasoning in SPEK.

Search for Similar Cases in SPEK. As described in reasoning process (2) of SPEK, similar transforms, that is, transforms whose LHS or RHS resembles a reaction site assumed on a given target, are sought in TKB. Figure 9 shows the manner of target-driven similar transform search, where TKB is structured and accessed from various aspects. While the entire TKB is organized hierarchically, not all the TKB layers are linearly ordered. Thus, a similar transform search is carried out by the following matching process in place of a simple maximal common subgraph matching.

(1) Extract a feature graph from a reaction site assumed on a target, which is represented as augmented EMN.

(2) Search for a transform in the feature TKB which matches the feature graph, where the augmented EMN is used as a key.

(3) In the case that the search is successful:

(i) If only one transform is found, it is evaluated as an applicable transform, that is, the favorability of the transform is estimated by consulting the yield value and reaction condition of its source instance.

(ii) If multiple transforms are found, they are pruned by expanding the matching area. Actually, the reaction site on the target is expanded to the maximal area defined by the algorithm described earlier to make a new EMN key, and the search is navigated to the maximal TKB. If only one transform is found in the maximal TKB, it is selected. If multiple transforms are found in the maximal TKB, they are pruned in the same manner as in the feature TKB. The pruning procedure is as follows. Suppose that $T = \{t_1, ..., t_n\}$ is the set of transforms found, where t_i: $p \rightarrow p_i$, $i = 1,...,n$. A transform $t_{\max} \in T$ that has a maximal common subgraph with the target is searched for. When the pruning is not sufficient, a transform with better favorability is preferred, which is estimated in terms of the yield and reaction conditions of its source instance.

(4) In the case that the search fails:

(i) The feature TKB is searched again for the same matching area with the following operations.

(a) Mask subordinate attributes: degp (degree in parent), ACT (atom class in transform), ring multiplicity, membership to rings, membership to functional groups, and ring multiplicity and then perform matching. In other words, matching is performed concerning ordinary EMN, not augmented EMN.

(b) Generalize functional groups whose "degp" attribute is 1, where generalization means the abstraction of functional groups into a generic class as well as unification of functional groups by FGX. The constraint degp = 1 means that the functional group in question is not a node which constitutes a skeletal structure of the parent graph of a matching area but a terminal node. Generation is performed using production rules which rewrite substructures into their superclasses and/or variables, provided that the substructures are registered in the semantic dictionary. A substructure rewritten as a variable is interpreted to match with any substructure. There is an approach to generalization where each generic class is regarded as a formal language, and the class to which the substructure inquestion belongs is determined by the language.[22,23] On the other hand, the approach employed here in SPEK provides rather simple production rules which express similarity, equivalence and translatability from the viewpoint of reactivity, rather than strict definition from the viewpoint of topological relationships. There are cases in which generic classes themselves are related in terms of, for example, hierarchy and mutual translation, and, thus, some functional groups belong to multiple generic classes. Figure 10 shows the manner of matching a reaction site using production rules and unification by FGI.

(c) Generalize the remaining functional groups if possible. Although this operation together with (b) is performed for a target, they correspond virtually to modification of similar transforms.

(ii) When the search fails for all those trials, the search condition is loosened. That is, the matching area is reduced by successively eliminating atoms from the exterior toward the interior. In fact, this treatment is the navigation from the feature TKB through the core TKB to theprimitive TKB. Furthermore, this is also a kind of generalization from the viewpoint of matching.

In an actual process in SPEK, the first stage after target analysis is a retrieval phase for RDB in which the target is sought whether or not it is matched with some product of reaction data. The second stage then is a transform search in refined TKB. The process described above follows those two stages. The manner of target-driven navigation in TKB and RDB is shown in Figures 5 and 9.

CONCLUSIONS

The methods of building and structuring TKB in SPEK and the manner of utilizing TKB in the reasoning process are described. While transforms in TKB are extracted automatically from RDB, except for those of refined TKB, the extracted reaction sites are not always appropriate. In fact, it seems that most transforms require some refinement. As a result, it becomes difficult to control combinatorial explosion of the retrosynthetic tree by means of simple implementation of RBR with the crude transform base, and the reliability of the reasoning itself inevitably decreases. In SPEK, information contained in source reaction data is not translated into a single rule (transform) but into multiple rules (transforms) which are layered hierarchically. The CBR mechanism is implemented with these TKB layers, including RDB. The access to transforms is carried out as a target-driven navigation in the TKB layers. While there have been several approaches to defining the similarity between chemical structures, the concept of a feature graph is defined in SPEK, which is an augmented subgraph representation of a reaction site. A feature graph is given as many attributes as possible, reflecting the viewpoint of reactivity being a topological attribute of atoms, which is a consequence of the assumption that transforms are extracted automatically from reaction instances containing only topological information. Furthermore, several supporting modules for automatic transform extraction are necessary to recognize and classify such substances as reagents, solvents, and catalysts which are often described as parts of starting materials or products, resulting in a multicomponent graph. Currently, the interactive mode of the reasoning function is realized in SPEK, where the stepwise evaluation of produced precursors is left to users. It is important to provide flexible representation organization of knowledge, i.e., transforms in this case, for the practical use of complex and large scale knowledge base systems.

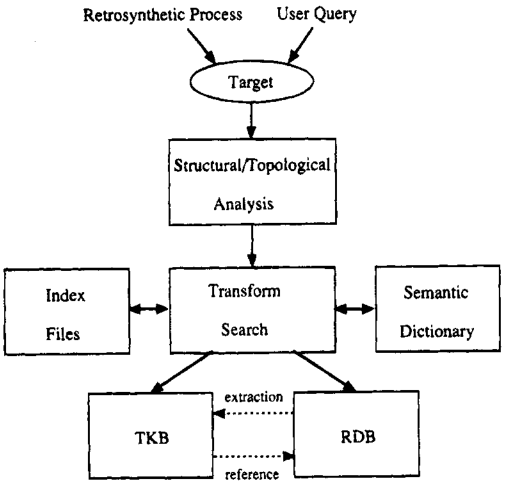


Figure 9. Target-driven access to TKB and RDB in SPEK. Targets are given by users or produced as precursors in retrosynthesis process. Search for similar transforms is carried out via feature analysis of targets. The index files and semantic dictionary are referred to as necessary in the search process. Navigation is allowed for any TKB and RDB.

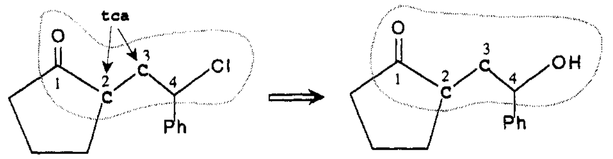


Figure 10. An example of unification by FGI. Assume that the left side is the target. The feature graph enclosed by a dotted line is extracted from the target, provided that atoms C2 and C3 are tca. Assuming that rewriting rules {Cl → hal, Cl → X} are provided, it then becomes possible to generalize Cl to hal using the rule Cl → hal in order to loosen the matching condition. However, if no similar transforms are found even by this generalization, then another rewriting rule Cl → X is applied to make atom Cl a variable. Since a variable is interpreted to be matchable with any substructure, the feature graph is matched with the 1−4 dioxygenation pattern, which becomes a subgoal. Then FGI is sought such that variable X is instantiated to atom O. In this case, FGI is found, and the first target is transformed to the precursor shown on the right.

ACKNOWLEDGMENT

The base of SPEK is given by RDB (reaction database). The author thanks Dr. Ikutoshi Matsuura of Chugai Pharmaceutical Co., Ltd., who designed the original representation of reaction data and provided tools for building RDB, and the members of the study group of reaction design, who have long been constructing RDB.

REFERENCES AND NOTES

(1) Corey, E. J.; Long, A. K.; Rubinstein, S. D. Computer-Assisted Analysis in Organic Synthesis, *Science*, **1985**, *228*, 408–418.

(2) Gelernter, H.; Rose, J. R.; Chen, C. Building and Refining a Knowledge Base for Synthetic Organic Chemistry via Methodology of Inductive and Deductive Machine Learning, *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 492–504.

(3) Lenat, D. B; Guha, R. V. Building Large Knowledge-Based Systems; Addison-Wesley: Reading Mass, 1990.

(4) Cutkosky, M. R.; Engelmore, R. S.; Fikes, R. E.; Genesereth, M. R.; Gruber, T. R. PACT: An Experiment in Integrating Concurrent Engineering Systems. *IEEE Computer* **1993**, *26*, 28–37.

(5) Neches, R.; Fikes, R.; Finin, T., Paril, R.; Senator, T.; Swartout, W. R. Enabling Technology for Knowledge Sharing. *AI Magazine* **1991**, *12*, 36–56.

(6) Blurock,E. S.: Computer-Aided Synthesis Design at RISC-Linz: Automatic Extraction and Use of Reaction Classes. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 505–510.

(7) Gasteiger, J.; Marsili, M.; Hutchings, M. G.; Saller, H.; Low, P.; Rose, P.; Rafeiner, K. Models for the Representation of Knowledge about Chemical Reactions. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 467–476.

(8) Matsuura, I. Development of PC-SYNTREX, A CAD System for Organic Synthetic Route, Based on Reaction Database. In *Proceedings of the 13th Symposium on Information Chemistry in Japan*; 1990, pp 17–20.

(9) Nakayama, T. Computer-Assisted Knowledge Acquisition System for Synthesis Planning. *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 495–503.

(10) Nakayama, T.: Semantic Dictionary for Substructure Matching of Chemical Structures with General Descriptors. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 845–853.

(11) Funatsu, K.; Endo, T.; Kotera, N.; Sasaki, S. Automatic Recognition of Reaction Site in Organic Chemical Reactions. *Tetrahedron Comput. Methodology* **1988**, *1*, 53–69.

(12) We do not deny the possibility that other systems implement the same kind of information.

(13) Warren, S. *Designing Organic Synthesis*; John Wiley & Sons: Chichester, 1978.

(14) Kobayashi, S. Present and Future of Case-Based Reasoning. *J. Japanese Society for Artificial Intelligence* **1992**, *7*, 3–9.

(15) Watanabe, H.; Okuda, K. A Method of Integrating Rule-Based Reasoning and Case-Based Reasoning. *IPSJ SIG Notes 9 4-AI-97*, **1994**, 11–20.

(16) Rajamoney, S. A.; Lee, H. U. Prototype-Based Reasoning: An Integrated Approach to Solving Large Novel Problems, In *Proceedings of Ninth National Conference on Artificial Intelligence, AAAI-91*; AAAI Press: 1991.

(17) Branting, L. K.; Porter, B. W. Rules and Precedents as Complementary Warrents, In *Proceedings of Ninth National Conference on Artificial Intelligence, AAAI-91*; AAAI Press: 1991.

(18) Golding, A. R.; Rosenbloom, P. S. Improving Rule-Based Systems through Case-Based Reasoning, In *Proceedings of Ninth National Conference on Artificial Intelligence, AAAI-91*; AAAI Press: 1991.

(19) Wochner, M.; Brandt, J; Scholley, A.; Ugi, I. Chemical Similarity, Chemical Distance, and its Exact Determination *CHIMIA* **1988**, *42*, 217–225.

(20) Willett, P; Winterman, V.; Bawden, D. Implementation of Nearest-Neighbor Searching in an Online Chemical Structure Search. *Chem. Inf. Comput. Sci.* **1986**, *26*, 36–41.

(21) Wilcox,C. S., & Levinson,R. A.: A Self-Organized Knowledge Base for Recall, Design, and Discovery in Organic Chemistry. In *Artificial Intelligence Applications in Chemistry*; Pierce, T. H., Hohne, B. A., Eds.; ACS Symp. Ser. 306; American Chemical Society: Washington, DC, 1986.

(22) Barnard, J. M.; Lynch, M. F.; Welford, S. M. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 2. GENSAL: A Formal Language for the Description of Generical Chemical Structures. *J. Chem. Inf. Comput. Sci.* **1981**, *21*, 151–161.

(23) Lynch, M. F.; Gillet, V. J.; Downs, G. M.; Holliday, J. D.; Barnard, J. M. Computer Storage and Retrieval of Generic Chemical Structures in Patents. 11. Theoretical Aspects of the Use of Structural Language in a Retrieval System *J. Chem. Inf. Comput. Sci.* **1991**, *31*, 233–253.

CI950045X