경영정보학연구
제7권 2호
1997년 9월

혼합 지식 기반 내 포함 이상의 검증: 메타 그라프적 접근**

이 선 로*

Verification of Subsumption Anomalies in Hybrid Knowledge Bases : A Meta-graph Approach

As object models and hybrid knowledge are increasingly used in current information systems development, Is-a structures need to be more formally defined, and subsequently subsumption related anomalies need to be detected with minimal declaration of meta knowledge. This paper extends a metagraph in the hybrid environments and demonstrates its utilities for detecting such anomalies that can occur from semantics and dynamics unique to the hybrid knowledge and data structure.

---

* 연세대학교 경영정보학과
** 본 연구는 연세대학교 교내 연구비로 수행되었음.

이 논문은 1997년 5월 15일 접수하여 1차 수정을 거쳐 1997년 11월 10일 게재확정되었습니다.

Ⅰ. Introduction

Knowledge bases (KB) are the primary targets of knowledge-based systems (KBS) management, since other components (e.g., inference engine, shell utilities, and external interfaces) are often reused with little change.

Previously, anomalies in rule bases have been characterized by consistency and completeness [Nazareth, 1988].

Such anomalies can generate invalid results and make maintenance more difficult due to unexpected side effects.

As the use of hybrid systems increases, verification methods and tools developed for rule-based systems (RBS) need to be modified for covering the anomalies unique to hybrid systems. Emphasizing object hierarchy and property inheritance, hybrid systems combine object-oriented and rule-based techniques.

High-end hybrid tools (i.e., KEE) are readily available and are being used for building large and complex KBS [Grabowski and Sanborn, 1994].

As noted in Lee and O'Keefe [1993], such tools still leave the burden of maintaining the frame structure and subsumption relationships among literals in rules entirely to developer.

In terms of verification, present efforts focus on rule-based systems, and little with few exceptions [Lee and O'Keefe, 1993] has been done in the context of hybrid systems environments. The objectives of this paper, therefore, are (1) to develop meta-graphic representations of hybrid knowledge, and (2) to provide a process for the dynamic analysis of the potential anomalies and knowledge simplification by removing redundant knowledge.

Ⅱ. A Review

There have been several intermediate methods, such as a dependency chart, a decision table, a graph, or Petri-net, proposed for representation of rule bases (RB) and identification of anomalies in rule-based systems (RBS).

Kuni [1994] represent rules in a form of decision trees and later transform it into decision tables to check consistency among rules. Kuni's approach, however, is limited to a static analysis of RBS. Valiente [1993] transforms the RB to be verified into its hypergraph representation [Berge, 1989] and apply a set of graph production (predefined graph production schemata) in order to check redundant and subsumed rules.

Their methods can provide a good visual walk-through the structure of rules. Valiente's method [1993] address the problem of global effects of redundant rules, even though it is just for removing static subsumption-related redundancy.

A numerical Petri-net has been used to overcome such a limitation as the static analysis of rule structures.

The reachability set of nets have been generated for dynamic analysis of inferencing in RB [Liu and Dillon 1991] and causal knowledge [Portinale, 1992].

Also, COVER system [Preece et al., 1992] employs a graph-based representation that allows for efficient checking over chains of inference, rather than comparing rules in pairs.

In short, Table 1 summerizes the characteristics of several studies that have utilized various methods for verification of target knowledge bases, and showed the different levels of transparency of inference and dynamics.

<Table 1> A comparison of previous approaches and meta-grahic one

	Method(s)	Target	Transparency	Dynamics
Kuni (1994)	Decision Tree/ Decision Table	Rules in RBS	Explicit sintax-based	Static/No Inference
Valiente (1993)	Hypergraph	Rules in RBS	Predfined global rule schema	Static/No Inference
Preece st al. (1992)	Gragh	Rules in RBS	Predefined metaknowledge	Static/Partially Dynamic
Portinale (1992)	Petri-net	Causal Knowledge	Explicit Reachability	Dynamic Inference
Liu and Dillon (1991)	Petri-net	Rules in RBS	Explicit Reachability	Dynamic Inference
This Study	Meta-graph	Hybrid Knowledge	Explicit and Implicit Reachablilty	Static/Dynamic Inference

Ⅲ. Meta-graphic Representation of Hybrid Knowledge

Metagraphs are graphical structure consisting of a set of elements and a set of edges that contain some of the elements [Basu and Blanning, 1994].

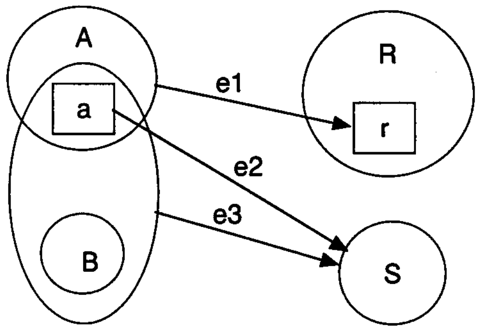
Each edge is an ordered pair of the invertex and outvertex, and at least one of these vertices must be non-null (a formal definition can be found in [Basu and Blanning, 1994]). Hybrid knowledge can be graphically represented as follows:

Circle : predicate symbol
Letter : object or entity name associated with predicate
Box : is-a relationship between (an object in a box is a kind of an object in a most adjacent circle)
Arrow : rule (model)

Now, suppose that there are three rules (r_1,r_2,r_3) as belows and five objects (A, a, B, R, S) associated with those rules. Further, let a and A be a ∈ A.

r_1: P(A) -> Q(r)
r_2: P(a) -> Q(S)
r_3: P(a) P(B) -> Q(S)

Then, this rule set can be graphically represented as follows:



Also, the above graph can be represented in an adjacent matrix, each cell of which may contain 4 elements[1]:

I: Coinput set: {}
O: By-product set {}
P: Path set <>
K: Is-a relation set [[x]|[y]]
[x]: subset or superset of objects associated with conditions
[y]: subset or superset of objects associated with actions,

and the resulting matrix for the above graph will be as follows:

	Q(r)	Q(S)
P(A)	$<0,0,<e_1>,[[a]\|[R]>$	
P(a)		$<0,0,<e_2>,[[A]\|0]>$
		$<\{B\},0,<e_3>,[[A]\|0]>$
P(B)		$<\{a\},0,<e_3>,[0\|0]>$

Further, for examining a connectivity among rules a closure matrix (A*) can be generated as described in Basu and Blanning [1994]. The matrix can be interpreted as (1) a set of column index: goals generated by a combination of different conditions, (2) a set of row index: hypothesis - a combination of condition(s), and (3) each entry: a tuple representation of a

1 Definitions of three elements (I, O, P) are equivalent to those of the metagraph [Basu and Blanning, 1994] developed for the flat knowledge structure, such as RBS. This study, however, has expanded their metagraph by adding the fourth element (K), and this extension allow us to explicitly represent object hierarchies that can cause anomalies unique to hybrid KBS.

hypothesis-goal implication.

For example, the entries of (P(a), Q(S)) indicates that there are two paths from P(a) to Q(S). Its first entry is associated with the path e_2 with no coinput and by-product. The difference between the first and the second entry is that the path e_3 included in the second entry has a coinput {B}.

The fourth element implies the object (a) used in a hypotheses has a parent object (A), but one in a goal does not have a hierarchical object structure.

Ⅳ. Subsumption Anomalies

Subsumption has been formally defined in Nilsson [1980] as follows: a clause Φ subsumes a clause Ψ if there exists a substitution σ such that $\Phi\sigma$ is a subset of Ψ, where $\Phi\sigma$ = a substitution instance of Φ using a substitution σ. This states that subsumption occurs whenever a clause after substitution is true, the superset of that clause is subsumed ($\Phi\sigma \rightarrow \Psi$).

In the traditional production systems, therefore, subsumption occurs in the case that the more general rule can always be fired whenever the restrictive rule is fired.

In a hybrid system, however, traditional subsumption cannot explain the hierarchical subsumption among objects when a clause contains inherited objects. The main difference between the traditional and hierarchical subsumption is explicitness of relationships among literals used in conditions and goals.

While occurrence of the traditional subsumption depends simply on the number of literals used in conditions and goals (e.g., conjunction of three literals is subsumed by conjunction of two literals), the hierarchical subsumption occurs because of the implicit inheritance relationship among objects used in literals (e.g., "I have a sports car" is hierarchically subsumed by "I have a car" since a car is more general than a sports car, but not vice versa. In this case a traditional subsumption does not occur because of the same number of literals with different constants used).

Lee and O'Keefe [1993] separate object constants from predicates and expand the scope of subsumption by explicitly stating the implicit inheritance among objects. Their study, however, has been limited to analysis of the static subsumption relations among propositions, and this study attempts to expand the static to dynamic analysis.

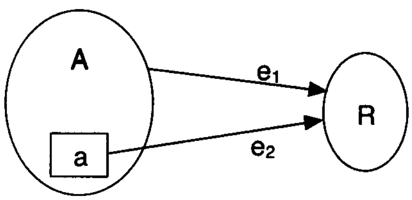
4.1. Complete subsumption

When every object constant used in a literal is a sub-class of every object constant used in the corresponding literal, conjunction of all subsuming literals completely subsumes conjunction of all subsumed literals. For example,

r_1: P(A) -> Q(R)

r_2: P(a) -> Q(R)

Rule 1 and 2 can be represented in both a graph and a matrix as follows.



	Q(r)
P(A)	$\langle 0, 0, \langle e_1 \rangle, [[a]\|0]\rangle$
P(a)	$\langle 0, 0, \langle e_2 \rangle, [[A]\|0]\rangle$

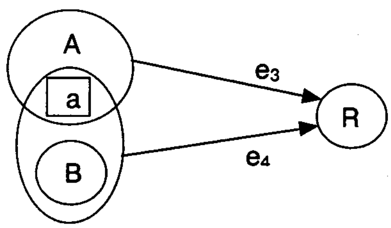
In the above matrix if $(|\cup I_{ij}|=|\cup I_{kj}|)(P_{ij} \neq P_{kj})(x_{ij} \in x_{kj})$ where $i \neq k$, then P_{ij} (e.g., e_1) completely subsumes P_{kj} (e.g., e_2), and e_1 can be considered as a redundant path (rule).

4.2. Partial subsumption

When a clause has more literals than its counterpart and there are common predicates between the two clauses, conjunction of more literals partially subsumes conjunction of fewer literals if object constants used in common predicates is a sub-class of its counterpart. For example[2],

r_3: P(A) -> Q(R)

r_4: P(a)P(B) -> Q(R)



	Q(R)
P(A)	$\langle 0, 0, \langle e_3 \rangle, [[a]\|0]\rangle$
P(a)	$\langle \{B\}, 0, \langle e_4 \rangle, [[A]\|0]\rangle$
P(B)	$\langle \{a\}, 0, \langle e_4 \rangle, 0\rangle$

If $(|\cup I_{ij}| \; |\cup I_{kj}|)$ $(x_{ij} \in I_{kj}$ under the

---

2 Through out the paper all examples show sample knowledge, its metagraph, and matrix representation in order.

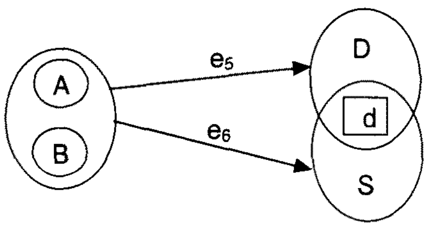
same P_{kj}) for $P_{ij} \neq P_{kj}$ where $i \neq k$, then P_{ij} (e.g., e_3) partially subsumes P_{kj} (e.g., e_4), and e_3 can be again considered as a redundant path (rule).

4.3. Semantic Conflict

In the production systems, two rules are in conflict if they succeed in the same constraints but with conflicting conclusions. In the hybrid systems, actions are often semantically (hierarchically) structured, and we have to understand the different meaning of semantically conflicting rules. In other words, semantic conflict occurs when a same set of conditions leads to conclusions in which objects used in goals are under the is-a relationships.

For example, following two rules show semantic conflict caused by complete subsumption among goals.

r_5: P(A) P(B) -> Q(D)
r_6: P(A) P(B) -> Q(d)

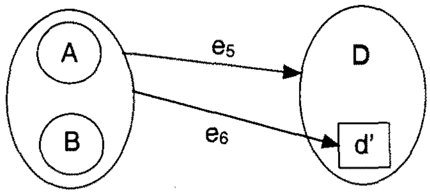


Q(D) Q(d)

P(A) <{B}, 0, <e_5>, [0|[d]]> <{B},0,<e_6>,[0|[D]]>
P(B) <{A}, 0, <e_5>, [0|[d]]> <{A},0,<e_6>,[0|[D]]>

Partial subsumption among goals also can lead to semantic conflict, as follows:

P(A) P(B) -> Q(D)
P(A) P(B) -> Q(d) Q(S)



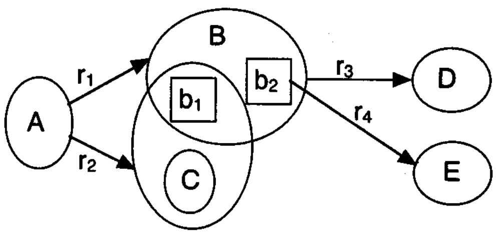
Q(D) Q(d) Q(S)
P(A) <{B},0,<e_5>,[0|[d]]> <{B},{S},<e_6>,[0|[D]]>
<{B},{d},<e_6>,[0|0]>
P(B) <{A},0,<e_5>,[0|[d]]> <{A},{S},<e_6>,[0|[D]]>
<{A},{d},<e_6>,[0|0]>

In the matrix, therefore, if ($I_{ij} = I_{ik}$ for $P_{ij} \neq P_{ik}$) ($y_{ij} \in \cup O_{ik}$) then P_{ij} (e.g., e_5) and P_{ik} (e.g., e_6) are in semantic conflict.

Although this semantic conflict does not always lead to contradictory actions, but it is an anomaly that should be examined particularly in case that such actions are defined as sub-goals, which can be chained with other rules. For example, the actions in r_1 and r_2 are in semantic conflict. Depending on the abstract level of the action(s) in r_1, therefore, the propagation among rules can be varied.

If $Q(b_3)$ instead of $Q(B)$ is derived from r_1, r_4 cannot be fired unless $Q(b_2)$ is predefined or derived from other rules. However, if $Q(B)$ is derived from r_1, r_4 should be fired since $Q(B)$ subsumes $Q(b_1)$. Hence, the side effects of rules should be examined when the abstract level of the action to be chained to other rules is determined.

r_1: P(A) -> Q(B)
r_2: P(A) -> $Q(b_1) \wedge Q(C)$
r_3: Q(B) -> S(D)
r_4: $Q(b_2)$ -> S(E)



A more serious impact of semantic conflict can occur when semantic conflict is associated with a logical negation as follows:

P(A) P(B) -> Q(R)
P(A) P(B) -> Q(r)

These two rules are in contradiction that cannot be checked through syntactic comparison.

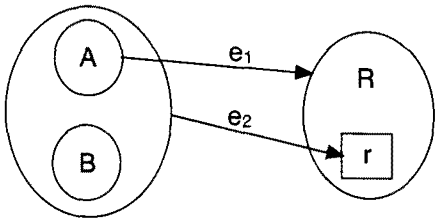
4.4. Structural Conflict (Type I)

When KB being constructed, a more general rule can be gradually refined in order to increase maintainability of KB. In other word, a more general rule subsumes a specific (refined) rule, and it can be called they are structurally in parallel. Depending on the inference chain among rules, two rules in structural parallel can be merged to improve performance or maintained as a form of the hierarchical refinement of rules.

Type I structural conflict, on the other hand, occurs when the condition set in the i^{th} rule subsumes the condition set in the j^{th} rule and the action set of the i^{th} rule is subsumed by the action set in the j^{th} rule. Depending on the type of subsumption among condition and action sets, we can consider three cases of such structural conflict:

Case I: The condition set in the i[th] rule is in the traditional subsumption relation with the condition set of the j[th] rule and the action set of the i[th] rule is hierarchically subsumed by the action set in the j[th] rule. For example,

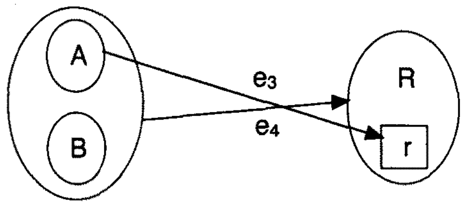
r_1:	P(A)	-> Q(R)
r_2:	P(A) P(B)	-> Q(r)



	R	r
A	<0,0,<e_1>,[0|[r]]>	<{B},0,<e_2>,[0|[R]]>
B		<{A},0,<e_2>,[0|[R]]>

r_1 and r_2 show structural parallel that can be identified by checking the matrix entries satisfying $(I_{ij} \in I_{ik}) \wedge (O_{ij}=O_{ik}=0) \wedge (P_{ij} \mp P_{ik}) \wedge (Y_{ij} \in Y_{ik})$. On the other hand, however, following example (r_3 and r_4) shows structural conflict.

r_3:	P(A)	-> Q(r)
r_4:	P(A) P(B)	-> Q(R)

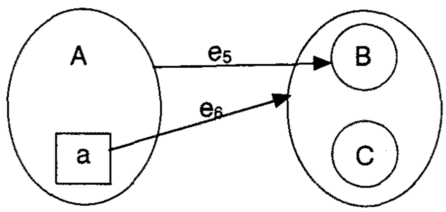


	R	r
A	<{B},0,<e_4>,[0|[r]]>	<0,0,<e_3>,[0|[R]]>
B	<{A},0,<e_4>,[0|[r]]>	

Such conflict can be represented with $((I_{ik} \in I_{ij}) \wedge (O_{ij}=O_{ik}=0) \wedge (P_{ij} \mp P_{ik}) \wedge (Y_{ij} \in Y_{ik}))$.

Case II: The condition set of the i[th] rule hierarchically subsumes the condition set in the j[th] rule, and the action set in the i[th] rule is in traditional subsumption relation with the action set of the j[th] rule. For example,

r_5:	P(A)	-> Q(B)
r_6:	P(a)	-> Q(B) Q(C)



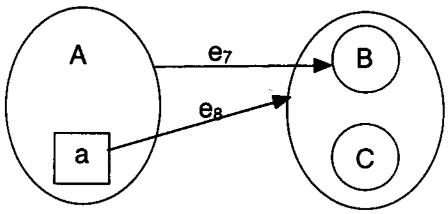
	B	C
A	<0,0,<e_5>,[[a]|0]>	
a	<0,{C},<e_6>,[[A]|0]>	<0,{B},<e_6>,[[A]|0]>

r_5 and r_6 show structural parallel satisfying $((O_{ij} \in O_{ik}) \wedge (P_{ij} \mp P_{ik}) \wedge (X_{ij}$

$\in X_{ik}$)). Again, however, following example (r_7 and r_8) shows structural conflict.

r_7: P(A) -> Q(B) Q(C)

r_8: P(a) -> Q(C)



B C

A <0,{C},<e7>,[[a]|0]>0,
<0,{B},<e7>,[[a]|0]>

a <0,0,<e8>,[[A]|0]>

In this case, therefore, it can be said that $((O_{ik} \in O_{ij}) \wedge (P_{ij} \neq P_{ik}) \wedge (X_{ij} \in X_{ik}))$ leads to structural conflict.

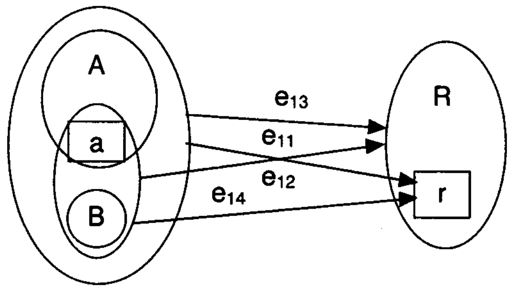
Case III: The condition set in the ith rule hierarchically subsumes the condition set of the jth rule and the action set of the ith rule is hierarchically subsumed by the action set in the jth rule. Consider the following rules:

R_{11}: P(A) P(B) -> Q(r)

R_{12}: P(a) P(B) -> Q(R)

R_{13}: P(A) P(B) -> Q(R)

R_{14}: P(a) P(B) -> Q(r)



r R

A <{B},0,<e11>,[[a]|[R]]> <{B},0,<e13>,[[a]|[r]]>

B <{A},0,<e11>,[0|[R]]> <{A},0,<e13>,[0|[r]]>

<{a},0,<e14>,[0|[R]]> <{a},0,<e12>,[0|[r]]>

a <{B},0,<e14>,[[A]|[R]]> <{B},0,<e12>,[[A]|[r]]>

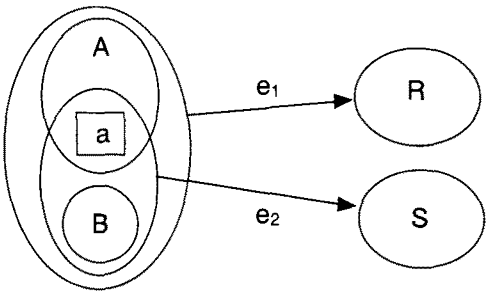
Structural parallel occurs when $(x_{ij} \neq 0) \wedge (y_{ij} \neq 0) \wedge (I_{ij} = I_{kl}) \wedge (O_{ij} = O_{kl}) \wedge (P_{ij} \neq P_{kl}) \wedge ((x_{ij} \in x_{kl} \in y_{ij} \in y_{kl}) \vee (x_{kl} \in x_{ij} \wedge y_{kl} \in y_{ij})$ for the row index that has the constant of x_{kl}). Hence, the paths e_{13} and e_{14} show an example of such structural parallel. Paths e_{11} and e_{12}, however, show structural conflict that occurs when $(x_{ij}$ $0)$ $(y_{ij}$ $0)$ $(I_{ij} = I_{kl})$ $(O_{ij} = O_{kl})$ $(P_{ij}$ $P_{kl})$ $((x_{ij}$ x_{kl} y_{kl} $y_{ij})$ $(x_{kl}$ x_{ij} y_{ij} $y_{kl})$ for the row index that has the constant of x_{kl}). The row and column indices can be generated for the diagonal comparisons using Eval function that converts the content of x_{ij} to the row index, and that of y_{ij} to the column index of the matrix. In this case, for example, $_{kl}$ = Eval(X_{ij},Y_{ij}).

4.5. Structural Conflict (Type II)

Structural conflict (type II) happens when two sets of conditions are in subsumption relation, but they lead to different conclusions among which there is no subsumption relationship. For example,

r_1: P(A) P(B) -> Q(R)

r_2: P(a) P(B) -> Q(S)

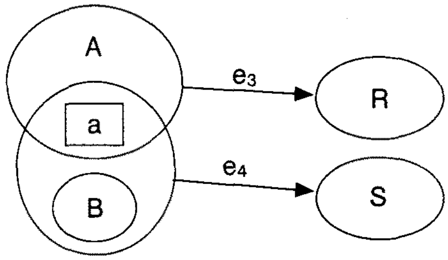


	Q(S)	Q(R)
P(A)		
<{B},0,<e_1>,[[a]|0]>		
P(a)	<{B},0,<e_2>,[[A]|0]>	
P(B)	<{a},0,<e_2>,0>	<{A},0,<e_1>,0>

For two rules that are in complete subsumption with the same number of predicates in their conditions, if $(P_{ij} \neq P_{ik}) \wedge (I_{ij} \in I_{ik}) \wedge (X_{ij} = X_{ik})$, then the condition set of P_{ij} subsumes that of P_{ik}. On the other hand, for those that are in partial subsumption with the different number of predicates in their conditions, however, if $x_{ij} \in (\cup I_{kl}$ under the same $P_{kl})$ for $P_{ij} \neq P_{kl}$, then the condition set of P_{ij} subsumes that of P_{kl}. For example,

r_3: P(A) -> Q(R)

r_4: P(a) P(B) -> Q(S)



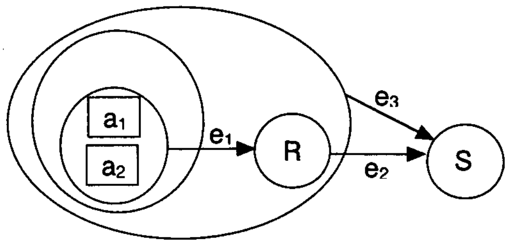
	Q(R)	Q(S)
P(A)	<0,0,<e_3>,[[a]|0]>	
P(a)		<{B},0,<e_4>,[[A]|0]>
P(B)		<{a},0,<e_4>,[0|0]>

For both cases if the corresponding goals (e.g., Q(R) and Q(S)) are in conflict, then it can be said that rule (r_3) is structurally conflicting (type II) with rule (r_4).

4.6. Chained subsumption

In hybrid systems chained subsumption can occur when an action in a rule is propagated to another rule as a condition and results in at least one subsumption relationship between a pair of literals in the condition clause. In such situations subsuming literal(s) become unnecessary. For example,

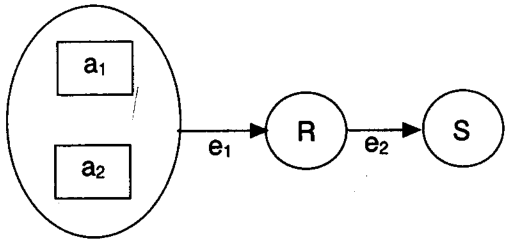
r_1: P(a_1) P(a_2) -> Q(R)
r_2: Q(R) -> U(S)
r_3: P(A) Q(R) -> U(S)



Since chained subsumption occurs when two or more rules are chained, they can be revealed by checking the transitive closure matrix as bellows:

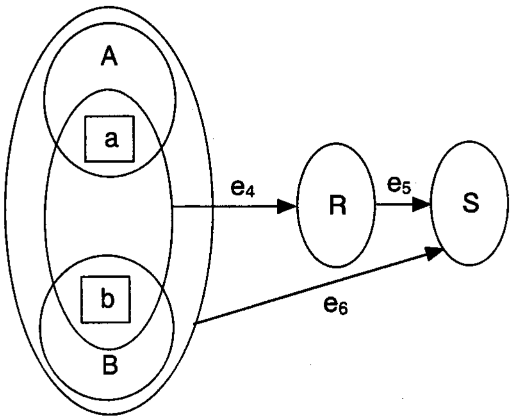
	Q(R)	U(S)
P(A)		<{R},0,<e_3>,[[a_1,a_2]|0]>
P(a_1)	<{a_2},0,<e_1>,[[A]|0]>	<{a_2},{R},<e_1,e_2>,[[A]|0]>
P(a_2)	<{a_1},0,<e_2>,[[A]|0]>	<{a_1},{R},<e_1,e_2>,[[A]|0]>
P(R)		<0,0,<e_2>,0>

If $(I_{ij} = O_{kj}) \wedge (O_{ij} \in O_{kj}) \wedge (X_{ij} \in X_{kj})$, then P_{ij} tends to include subsuming literal(s) (e.g., P(A)) which can be deleted. Subsequently, r_2 and r_3 become logically equivalent, and the above graph can be simplified as:



In this case the resulting rule set after forward-chained show partial subsumption in condition set. After r_1 is forward-chained to r_2, for example, the condition sets of r_2 and r_3 are in partial subsumption. On the other hand, the resulting rule set after forward-chained can have complete subsumption in condition set as shown in the following example.

r_4: P(a) P(b) -> Q(R)
r_5: Q(R) -> U(S)
r_6: P(A) P(B) -> U(S)



Again, corresponding closure matrix will be:

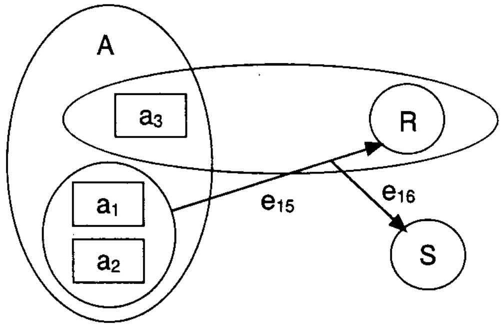
	Q(R)	U(S)
P(a)	<{b},0,<e_4>,[[A]|0]>	<{b},{R},<e_4,e_5>,[[A]|0]>
P(b)	<{a},0,<e_4>,[[B]|0]>	<{a},{R},<e_4,e_5>,[[B]|0]>
Q(R)	<0,0,<e_5>,0>	
P(A)	<{B},0,<e_6>,[[a]|0]>	
P(B)	<{A},0,<e_6>,[[b]|0]>	

After forward-chained, the condition sets of r_5 and r_6 are in complete subsumption, and this kind of chained complete subsumption can be represented with $((I_{ij} \in I_{kj}) \wedge (P_{kj} \cap p_{ij}=0) \wedge (X_{kj} \in X_{ij})$ for the same p_{ij} and the same p_{kj}). Subsequently, P_{kj} tends to include subsuming literal(s) (e.g., $P(A) \wedge P(B)$), which subsumes P_{ij} and becomes redundant.

4.7. Unreachable Goals

When objects that have the same parents are bound with a variable in the conditions, the rule will not be fired unless the variable is allowed to have multiple values. For example, consider the following rules.

r_1: $P(a_1)$ $P(a_2)$ -> Q(R)
r_2: $P(a_3)$ Q(R) -> Q(S)



	Q(R)	Q(S)
$P(a_1)$	<{a_2},0,<e_{15}>,[[A]|0]>	
$P(a_2)$	<{a_1},0,<e_{15}>,[[A]|0]>	
$P(a_3)$		<{R},0,<e_{16}>,[[A]|0]>

Based on the entries of the matrix, a goal (e.g., Q(R)) may not be reachable when $(X_{ij}=X_{kj})$ $(Y_{ij}=Y_{kj}) \wedge (I_{ij} \in X_{ij}) \wedge (I_{kj} \in X_{kj})$ for $P_{ij}=P_{kj}$. However, if I_{ij} and I_{kj} are not mutually exclusive, then the goal can be reached. For example, an undergraduate student can also be a president of a student union, but cannot be a graduate student in a same school.

It is not so obvious, however, to detect the AND conflict which can occur when a consequent becomes a part of conditions of other rules. For example, Q(R) in r_1 will be forward-chained to rule 2 as a condition and the two rules now have constants which are mutually exclusive with a common parent.

Consequently, Q(S) will become an unreachable goal. A meta-predicate (Stachowitz and Coombs, 1987) or impermissible set (Preece and Shinghal, 1991) (e.g., INCOMPATABLE (male(x), female(x))) have been predefined to detect unreachable goals. The transitive closure matrix for the above example is:

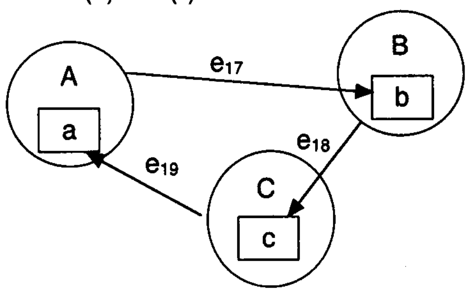
	Q(R)	Q(S)
$P(a_1)$		$<\{a_2\},0,<e_{15}>,[[A]\mid 0]>$
$<\{a_2\},\{R\},<e_{15},e_{16}>,[[A]\mid 0]>$		
$P(a_2)$		$<\{a_1\},0,<e_{15}>,[[A]\mid 0]>$
$<\{a_1\},\{R\},<e_{15},e_{16}>,[[A]\mid 0]>$		
$P(a_3)$		$<\{R\},0,<e_{16}>,[[A]\mid 0]>$

and a goal Q(S) will not be reachable when $(O_{ij}=I_{kj}) \wedge (X_{ij}=X_{kj}) \wedge (P_{kj}P_{ij})$ INCOMPATABLE$(I_{ij},X_{ij}-I_{ij})$.

4.8. Circularity

Traditionally, a set of rules is circular if the chaining of those rules in the set forms a cycle, and such a system will enter an infinite chain at run time. Rules are connected when at least one literal in action statements is syntactically matched with at least on literal in condition statements. For example, consider the following rule set:

P(A) -> Q(b)
Q(B) -> R(c)
R(C) -> P(a)



	Q(a)	R(b)	P(c)
P(A)		$<0,0,<e_{17}>,[[a]\mid[B]]>$	
Q(B)			$<0,0,<e_{18}>,[[b]\mid[C]]>$
R(C)	$<0,0,<e_{19}>,[[c]\mid[A]]>$		

Since Q(b) and R(c) are subsumed by Q(B) and R(C) respectively, the above rule set is circular when forward-chained.

In the case of backward-chaining, given a goal of P(a), the rule set will be circular when P(a) is explicitly in a database or derived before the goal is given since the subgoal of P(a), including Q(B) and R(C), are subsequently satisfied with an implication of P(A).

In order to overcome the limitation of syntactic pattern match, when ($y_{ij} \neq$ nil) $\wedge$ (the object constant in a column index $\in y_{ij}$), y_{ij} can be added to O_{ij} since a certain combination of conditions leads to a same types of goals (e.g., y_{ij} subsumes the object constant in a column index).

Subsequently, this operation will generate a following matrix:

	Q(a)	Q(b)	Q(c)
A		$<0, \{\mathbf{B}\},<e_{17}>,[[a]\|[B]]>$	
B			$<0,\{\mathbf{C}\},<e_{18}>,[[b]\|[C]>$
C	$<0,\{\mathbf{A}\},<e_{19}>,[[c]\|[A]>$		

Then, the transitive closure operation can be applied for this transformed adjacent matrix. For example,

A to c: $<0,\{b,B\},<e_{17},e_{18}>,[[a]|[C]]>$

A to a: $<0,\{b,B,c,C\},<e_{17},e_{18},e_{19}>,[[a]|[A]>$

Traditionally, $X_{ij} = Y_{ij}$ or $X_{ij} = Y_{ij} = 0$ means circularity. In the hybrid system, however, such circularity is implicit under the subsumption relationships among object constants associated with predicates in hypothesis. Now, if $X_{ij} \subseteq Y_{ij}$ in a transitive closure matrix, then P_{ij} is a *complete* circular path. If $y_{ij} \in x_{ij}$ in a transitive closure matrix, then P_{ij} is a *partial* circular path that may not be circular depending on the facts defined or derived.

Unlike the traditional static approach, as shown in chained subsumption, unreachable goals, and circularity, the transitive closure matrix representation of KB allows dynamic analysis among rules.

V. Verification Process

The procedure for identifying subsumption anomalies caused by structural semantics, using meta-graphic transformation, can be character- ized by following steps:

Step 1 : convert hybrid KB to meta-graphs.

Step 2 : convert the meta-graph to adjacent matrices.

Step 3 : conduct static analysis.
- Intra-cell operation
- Column operation
- Row operation
- Diagonal operation

Step 4 : construct A* matrix.

Step 5 : conduct dynamic analysis.
- Chained subsumption analysis
- Circularity analysis

In this section, each verification step will be illustrated using the following hybrid KB example.

<u>An Hybrid KB Example</u>

For the sake of simplicity, let objects represented with capital letters are parents and those with small letters are children associated with corresponding capitals.

r_1: A -> d
r_2: a -> D
r_3: $a \wedge B$-> D
r_4: $B \wedge C$-> F
r_5: $b \wedge C$-> E
r_6: $A \wedge B \wedge C$-> e
r_7: $A \wedge B \wedge C$-> f G
r_8: $A \wedge B \wedge C$-> F
r_9: $c_1 \wedge c_2$ -> F
r_{10}: D -> H
r_{11}: E -> F
r_{12}: $F \wedge c_3$ -> I
r_{13}: $C \wedge F$ -> I
r_{14}: H -> A
r_{15}: $e \wedge G$ -> F

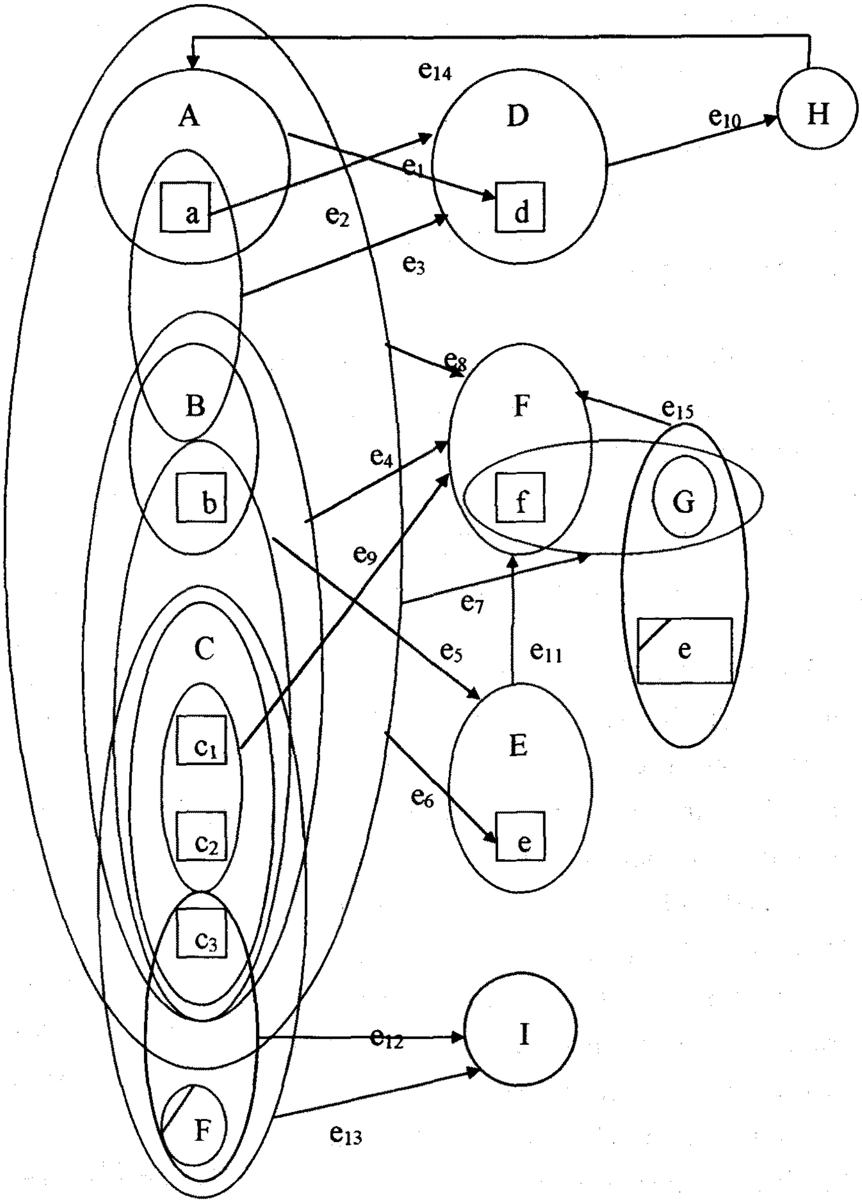


Figure 1: A Meta-Graph for the Sample KB[3]

3 The slashed circle or rectangle means duplicated one for simplication.

A meta-graph that reflects the knowledge structure of the above KB example can be as shown in Figure 1. The adjacent matrix (M) can be generated as in Table 2[4]. Null column entry means that the predicate used as column index is a given fact, while null row entry indicates that the predicate used as row index is a goal. The predicate used non-null column and row index can be interpreted as a sub-goal.

Table 2: The adjacent matrix and partial result of A* for the sample KB.

Goal / Condition	A	D	d	E	e	F	f	G	H	I
*A	$<0,\{d,b\},<e_1,e_{10},e_{14}>,[a/a]>$		$<0,\{D\},<e_1>,[[a]\mid[D]]>$		$<\{B,C\},0,<e_6>,[[a]\mid[E]]>$	$<\{B,C\},0,<e_8>,[[a]\mid[f]]>$	$<\{B,e\},\{G\},<e_7>,[[a]\mid[F]]>$	$<\{B,C\},\{f\},<e_7>,[[a]\mid 0]>$		
a		$<0,0,<e_2>,[[A]\mid[d]]>$ $<\{B\},0,<e_3>,[[A]\mid[d]]>$								
B		$<\{a\},0,<e_3>,[[b]\mid[d]]>$			$<\{A,C\},0,<e_6>,[[b]\mid[E]]>$	$<\{C\},0,<e_8>,[[b]\mid[f]]>$ $<\{A,C\},0,<e_8>,[[b]\mid[f]]>$	$<\{A,C\},\{G\},<e_7>,[[b]\mid[F]]>$	$<\{A,C\},\{f\},<e_7>,[[b]\mid 0]>$		
b				$<\{c\},0,<e_5>,[[B]\mid[e]]>$		$<\{C\},\{E\},<e_5,e_{11}>,[[B]\mid[f]]>$				
C				$<\{b\},0,<e_5>,[[c_1c_2c_3]\mid[e]]>$	$<\{A,B\},0,<e_6>,[[c_1c_2c_3]\mid[E]]>$	$<\{B\},0,<e_4>,[[c_1c_2c_3]\mid[f]]>$ $<\{A,B\},0,<e_8>,[[c_1c_2c_3]\mid[f]]>$	$<\{A,B\},\{G\},<e_7>,[[c_1c_2c_3]\mid[F]]>$	$<\{A,B\},\{f\},<e_7>,[[c_1c_2c_3]\mid 0]>$		$<\{F\},0,<e_{13}>,[[c_1c_2c_3]\mid 0]>$
C1						$<\{C_2\},0,<e_9>,[[C]\mid[f]]>$				$<\{C_2\},\{F\},<e_9,e_{12}>,[[c]\mid 0]>$
C2						$<\{C_1\},0,<e_9>,[[C]\mid[f]]>$				$<\{C_1\},\{F\},<e_9,e_{12}>,[[c]\mid 0]>$
C3										$<\{F\},0,<e_{12}>,[[c]\mid 0]>$
D									$<0,0,<e_{10}>,[[d]\mid 0]>$	
d										
E						$<0,0,<e_{11}>,[[e]\mid[f]]>$				
F										$<\{C_1\},0,<e_{12}>[[f]\mid 0]>$ $<\{C\},0,<e_{13}>[[f]\mid 0]>$
H	$<0,0<e_{14}>,[0\mid[a]]>$									
e						$<\{G\},0,<e_{15}>,[[E]\mid[f]]>$				
G						$<\{e\},0,<e_{15}>,[[0]\mid[f]]>$				

4 *Italics* indicate the result of A* operation explained later in this section.

Step 3 includes four operations for static analysis between paths that have a length of one (e.g., $|P_{ij}| = 1$). The intra-cell operation indicates that rules (paths) associated with a certain row and column index are examined in a cell by cell manner. The contents of M(aD), for example, contains two paths (e_2 and e_3) show the traditional condition subsumption and e_2 becomes a redundant path. The path (e_2), therefore, can be entirely eliminated from the matrix. Also, complete subsumption through at least one common literal in conditions, however, can be detected through the intra-cell operation.

The row operation means that the paths of the varying row index under the fixed column index. In the above example, The paths (e_{11} and e_{15}) of M(EF), M(eF), and M(GF) reveal the partial subsumption, and e_{11} can be eliminated as long as a fact (e) has been derived or defined in the knowledge base. Further, the paths (e_{12} and e_{13}) of M(CI) and M(c_3I) reveal the hierarchical complete subsumption.

Also, the goal (F) to be generated through the path (e_9) defined in M(c_1F) and M(c_2F) becomes unreachable if c_1 and c_2 are mutually exclusive.

Such hierarchical complete/partial subsumption and unreachable goals can be identified through the row operation.

The column operation indicates that the paths of the varying column index under the fixed row index.

The traditional goal subsumption, semantic conflict, and structural conflict (II) can be detected through this column operation.

The contents of M(AF) and M(Af), for example, include the paths (e_7 and e_8) that are in semantic conflict. Also, the paths (e_4 and e_5) defined in M(CF) and M(CE) respectively shows structural conflict (II).

The diagonal operation can be performed on varying row and column index for detecting structural conflict (I). But, a necessary condition for the diagonal operation is $X_{ij} \neq 0 \wedge Y_{ij} \neq 0$, which means the objects used in conditions and goals have a hierarchical structure. For example, M(Ad) and M(aD) include paths e_1 and e_2 that are in type I structural conflict.

The graph and matrix can be

simplified by eliminating redundant paths through the static analysis. Now, based on the simplified matrix, A* can be generated for dynamic analysis by propagating subsumption relations among objects in the corresponding subgoals as explained in section 4.8.

Then the chained subsumption, unreachable goals, and circularity anomalies can be identified. For example, A*(bF) includes a path <e_5,e_{11}> which indicates that r_5 is forward-chained to r_{11}. The resulting conditions after forward-chained are completely subsumed by those of r_4, and r_4 becomes redundant. Also, A*(c_1I) and A*(c_2I) indicate r_9 is forward-chained to r_{12} represented with A*(FI), and it is necessary to examine an unreachable goal by checking compatibility among literals in the resulting condition set.

Circularity can be checked by examining Is-a relation set. For example, X_{AA}=Y_{AA} in A*(AA) indicates that the paths <e_1,e_{10},e_{14}> become circular.

As described above, rather than all cells compared in the brute force way, the cells satisfying predefined conditions for certain anomalies will selectively be compared.

Further, the verification process should be interactively performed by knowledge engineers providing the verification processor with their opinions for clarification, rather than totally automated. As each step in the process is interactively performed, the KB can be progressively simplified by removing anomaleous knowledge, and the search space will be correspondingly reduced. Consequently, computational overhead can be minimized at each stage.

VI. Conclusions

Object-oriented approach has been advanced in many areas, such as systems analysis and design, data modeling, and knowledge engineering. However, Is-a structure needs to be more formally defined [Goldstein and Storey,1992], and such formalism can lead to logical verification of object models.

This paper extends the metagraph in the hybrid environments and applies it to analyze subsumption and its related anomalies that has been identified [Lee and O'Keefe, 1993] as a critical problem for improving the consistency and completeness of hybrid knowledge bases. Unlike previous research mostly limited to verification of the static rule structure and explicit inference among rules in the RBS as shown in Table 1, this meta-graphic approach addresses the issues of the implicit semantics in object hierarchies and the dynamics of inference in the hybrid knowledge structure.

⟨REFERENCES⟩

Basu, A. and Blanning, R.W., "Metagragh: a tool for modeling decision support systems," to appear in *Management Science.*

Berge, C., *Hypergraphs,* North-Holland, Amsterdam, 1989.

Goldstein, R.C. and Storey, V.C., "Unravelling Is-a structure," *Information Systems Research,* 3, 2, 1992, pp. 99-126.

Grabowski, M.R. and Sanborn, "Knowledge- representation and reasoning in a real time operational control: the shipboard piloting expert system (SPES)," *Decision Sciences,* 23, 6, 1992, pp. 1277-1296.

Higa, K., "Quality measurement of rule bases," *Proceedings of the Third Annual International Management Review,* Vol.35, No.2, Winter 1994, pp. 73-87.

Association of Knowledge Engineers Symposium, Washington D.C. November 1922.

Lee, S. and O'Keefe, R.M.,"Subsumption anomalies in hybrid knowledge bases," *International Journal of Expert Systems,* 6, 3, 1993, pp. 299-320.

Liu, L.K. and Dillon, T., "An approach towards the verification of expert systems using numerical petri nets," *International Journal of Intelligent Systems,* 6, 1991, pp. 255-276.

Nazareth, D.L., "An analysis of techniques for verification of logical correctness in rule-based systems," Unpublished dissertation, Department of Managerial Studies, Case Western Reserve University, 1988.

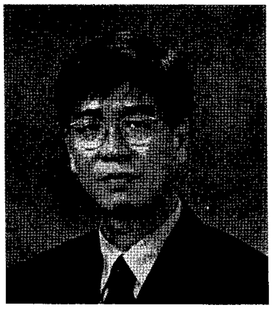
Nilsson, N.J., *Principles of Artificial Intelligence*, Los Altos, CA: Morgan Kaufmann, 1980.

Portinale, L., "Verification of causal models using petri nets," *International Journal of Intelligent Systems*, 7, 1992, pp. 715-742.

Preece, A.D., Shinghal, R., and Batarekh, A.,"Verifying expert systems: a logical framework and a practical tool," *Expert Systems With Applications*, 3, 3, 1992, pp. 421-436

Valentine, G., "Verification of knowledge base redundancy and subsumption using graph transformation," *International Journal of Expert systems*, 6, 3, 1993, pp. 341-355.

◆ 저자소개 ◆



이 선 로 (Lee, Sun Ro)
Rensselaer Polytechnic Institute(RPI)에서 경영정보학 박사학위를 취득하고 홍콩과학기술대학에서 조교수로 재직한 바 있으며 현재 연세대학교 경영정보학과 학과장으로 재직하고 있다. 주요 연구분야는 정보시스템 계획 및 평가, 시스템 개발 생산성 측정, 정보자원관리 등에 있으며 관련 논문 20여 편을 IEEE Transaction 학술지를 포함하여 해외 주요 학술지에 발표해 오고 있다. 현재 정부기관과 민간 기업에서 지원을 받아 의료분야와 섬유산업 분야의 ERP 템플레이트를 구축하는 프로젝트를 수행하고 있다.