

論文2000-37CI-3-2

공간 데이터베이스를 위한 새로운 위상 관계 유도 알고리즘

(A New Algorithm for Deriving Topological Relationships in Spatial Databases)

黃 煥 圭 *

(Whan-Kyu Whang)

요 약

위상 관계는 공간 데이터베이스의 질의 최적화에 중요한 역할을 한다. 만일 위상 관계를 사전에 알 수 있다면 비용이 많이 드는 질의 처리는 피할 수 있다. 본 논문에서는 1) 공간 객체간의 위상 관계의 파악, 2) 주어진 위상 관계의 일관성 검사, 3) 주어진 위상 관계로부터 새로운 위상 관계의 유도 등의 문제에 관하여 논의하고자 한다. 이러한 논의는 위상 관계와 관련된 질의가 주어졌을 때 효과적인 공간 질의 처리를 가능하게 하여준다. 특히 위상 관계로서 8가지 종류(equal, disjoint, overlap, meets, contains, contained-in, properly-contains, proerly-contained-in)를 고려한다. 본 논문에서는 주어진 위상 관계의 일관성을 검증하는 알고리즘을 제시하고 주어진 위상 관계로부터 모든 가능한 새로운 위상 관계를 유도하는 알고리즘을 제시한다.

Abstract

Topological relationships play an important role in query optimization in spatial databases. If topological relationships are known a priori, then expensive query processing can be avoided. In this paper we address the problems of: 1) identifying topological relationships among spatial objects, ii) checking consistency of specified topological relationships, and iii) exhaustively deriving new topological relationships from the ones specified. These activities lead to an efficient query processing when queries associated with topological relationships are invoked. Specifically, eight types of topological relationships ((equal, disjoint, overlap, meets, contains, contained-in, properly-contains, and properly-contained-in)) are considered. We present an algorithm to check the consistency of specified topological relationships and to derive all possible relationships from the given set of known relationships.

I. 서 론

* 正會員, 江原大學校 情報通信工學科
(Dept. of Information and Telecomm. Eng.,
Kangwon National University)

※ 본 연구는 97-99년도 정보통신연구진흥원의 대학기초연구사업의 부분적인 지원과 정보통신부의 정보통신 우수지법학교지원 사업의 부분적인 지원에 의한 결과임.

接受日字:1998年11月 19日, 수정완료일: 2000年 4月10日

최근 지리 정보 시스템(geographic information systems: GIS), CAD/CAM, 컴퓨터 비전, 로봇틱스, 지도 제작, 이미지 처리 등 일반 객체와 공간 객체(spatial object, 이하 간략히 객체라 표현)를 동시에 다루는 응용 분야가 점차 확대됨에 따라 공간 데이터베이스(spatial database)에 대한 관심이 크게 증가하고 있다^{[6][8]}. 이러한 새로운 응용 분야에서는 점, 선, 사각형, 다각형 등의 기본 객체와 여러 기본 객체들로 구성되는 복합 객체 등을 다루게 된다. 지금까지 공간

데이터베이스 분야에 관한 연구는 주로 복합 객체를 위한 공간 데이터 모델링과 각 객체를 보다 효과적으로 액세스하기 위한 공간 액세스 기법에 초점이 맞추어져 왔다^[6]. 또한, 공간 질의 처리 및 최적화에 관해서는 많은 연구들이 공간 액세스 기법을 이용한 공간 조인 문제를 다루고 있다^[3, 7, 10].

공간 데이터베이스에서는 객체간의 관계에 대한 질의가 빈번하게 사용된다. 예를 들어, “주어진 한 국가와 인접한 모든 국가들을 검색하라,” “주어진 한 객체와 교차하는 모든 객체를 검색하라” 등은 GIS와 같은 공간 데이터베이스 응용 분야에서 자주 나타나는 질의이다. 지금까지 이러한 질의는 주로 공간 액세스 기법에서 제공하는 인덱스를 탐색함으로써 처리할 수 있었다. 그러나 객체간의 공간 관계를 사전에 미리 파악할 수 있다면, 공간 관계에 관련된 질의들을 처리하기 위한 오버헤드를 크게 줄일 수 있다^[5, 13].

공간 관계는 크게 방향 관계(directional relationship), 거리 관계(distance relationship), 위상 관계(topological relationship)로 분류된다^[12]. 방향 관계의 예로는 left-of, right-of, above, below, in-front-of, behind 등이 있으며, 거리 관계의 예로는 near, far 등이 있다. 또한, 위상 관계의 예로는 equal, disjoint, overlap, meet, contain, contained-in, properly-contain, properly-contained-in 등이 있다. 이들 중 가장 기본적인 공간 관계는 위상 관계이며, 현재 이에 관한 몇몇 연구가 수행되어 왔다^[4,5,13]. 본 논문에서는 이중에서 위상 관계에 대한 추론에 논의의 초점을 맞추고자 한다.

본 연구에서는 전체 객체에 관한 위상 관계 정보가 사전에 주어짐을 가정한다. 위상 관계는 주어진 객체들의 기하학적인 위치 정보를 기반으로 하여 수동적 혹은 자동적으로 수집할 수 있으므로 이러한 가정은 자연스러운 것이다. 주어진 객체 집합으로부터의 위상 관계 정보는 계층 그리드 파일(multilevel grid file)^[19], 그리드 파일(grid file)^[11], 쿼드 트리(quad tree)^[15], K-D 트리(K-D tree)^[2], K-D-B 트리(K-D-B tree)^[14], R 트리(R tree)^[9], R+ 트리(R+ tree)^[17], R* 트리(R* tree)^[1] 등 기반이 되는 공간 액세스 기법을 이용하여 수집할 수 있으며, 수집 알고리즘은 저장 구조에 따라 차이가 있다.

이러한 자동 수집을 위한 수행 오버헤드는 매우 크며, 따라서 완전한 위상 관계 집합을 구한다는 것은

쉽지 않은 일이다. 객체의 수가 n 개인 경우 임의의 두 객체간의 가능한 모든 위상 관계의 수는 $nC_2 = n*(n-1)/2$ 가 되므로 객체의 수가 작은 경우에도 가능한 위상 관계의 수는 매우 커진다. 예를 들면, 10,000개의 객체들이 존재하는 경우 고려해야 할 위상 관계의 수는 49,995,000이 된다. 따라서 사전에 주어지는 위상 관계 정보는 완전한 것이 아닌 핵심이 되는 일부분의 것인 경우가 대부분이다.

본 논문에서는 사전에 부분적으로 주어진 위상 관계들간의 일관성을 검증하는 문제와 이로부터 새로운 위상 관계들을 추론하는 문제에 관하여 논의하고자 한다. 두 객체간의 위상 관계가 명시되어 있지 않는 경우에도 주어진 다른 위상 관계들을 기반으로 이들의 관계를 추론할 수 있다. 예를 들어, 객체 X가 객체 Y에 포함된다는(contained-in) 위상 관계, 객체 Y가 객체 Z에 포함된다는 위상 관계가 주어진다면, 객체 X는 객체 Z에도 포함된다는 새로운 위상 관계를 추론해 낼 수 있다. 본 논문에서는 이와 같이 부분적으로 주어진 위상 관계 집합으로부터 새로운 위상 관계를 추론할 수 있는 알고리즘을 고안한다. 새로운 위상 관계의 올바른 추론은 주어진 위상 관계 집합 내에서 상호 모순이 없는 경우에 한하여 가능하다. 따라서 본 논문에서는 이를 파악하기 위한 일관성 검증 알고리즘도 아울러 고안한다. 이러한 일관성 검증과 새로운 위상 관계의 추론은 같은 시점에서 요구되므로 본 논문에서는 이 두 가지 기능을 하나의 알고리즘에 통합한 단일 알고리즘을 제시한다.

본 논문과 관련된 논문으로 Sistla^[18] 등이 제시한 방법이 있으며 이것은 기본적인 공간 연산자에 대한 추론 규칙을 제공하는 것으로서 이론에 치중하였다. 공간 관계는 left-of, right-of, behind, in-front-of, above, below, inside, outside, overlaps 등으로 이것은 위상 관계와 방향 관계의 일부에 국한하여 규칙을 유도하였다. 본 논문에서 제시한 알고리즘은 8가지의 모든 위상 관계를 고려하였으며, 추론 결과는 이행 규칙 테이블(transitive rule table)이 하나의 결과를 나타내지 않을 때는 보다 구체적으로 가능한 값을 제공해 준다. 따라서 본 논문은 모든 위상 관계를 고려하였기 때문에 위상 관계에 관한 추론으로서 완전(complete)하며, 추론의 결과가 하나의 관계를 나타내지 않을 때 가능한 관계를 제시함으로써 8가지의 가능한 관계 중에서 보다 한정된 관계를 얻을 수 있는 장

점이 있다.

본 논문의 구성은 다음과 같다. 먼저, 제 2장에서는 객체들간의 위상 관계를 정의하고, 새로운 위상 관계 추론을 위한 이행 규칙 테이블(transitive rule table)에 관하여 설명한다. 제 3장에서는 부분적으로 주어진 위상 관계들간의 일관성을 검증하고, 이들을 기반으로 새로운 모든 위상 관계들을 추론하는 통합된 알고리즘을 제시한다. 제 4장에서는 구현 이슈로서 제안된 알고리즘을 Prolog언어를 이용하여 구현하는 방안에 관하여 다룬다. 끝으로 제 5장에서는 결론을 내리고, 향후 연구 방향에 관하여 논의한다.

II. 위상 관계 및 이행 규칙 테이블

본 장에서는 관련 연구로서 객체간의 위상 관계에 대하여 설명하고, 두 위상 관계를 이용하여 새로운 위상 관계를 유도하기 위한 규칙들을 정의한 추리 규칙 테이블에 관하여 기술한다.

위상 관계는 공간상에 객체 상호간의 위치 관계를 의미하며, 두 객체 X와 Y간의 위상 관계는 X의 경계선(boundary), 내부(interior), 외부(exterior)와 Y의 경계선, 내부, 외부와의 각각의 관계에서 나타나는 아홉 가지 교집합에 의하여 결정되며, 각 교집합이 공집

합인지 아닌지의 여부에 따라 29(=512)개의 서로 다른 위상 관계가 존재한다^[8]. 그러나 이러한 512개의 위상 관계들 중 실제로 발생 가능한 것은 단 여덟 가지이며, 이들은 그림 1에 나타난 바와 같이 equal, disjoint, overlap, meet, contain, contained-in, properly-contain, properly contained-in 이다.

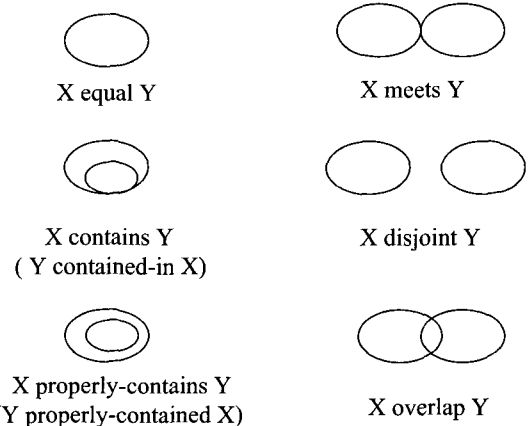


그림 1. 공간 데이터베이스에서 가능한 위상 관계 집합 Fig. 1. Topological relationships in spatial databases.

이들 중 contain과 contained-in, 그리고 properly contain과 properly contained-in은 각각 서로의 역

표 1. 위상 관계를 위한 이행 규칙 테이블 Table 1. Transitive Rule Table in Topological Relationships.

Y R ₂ Z \ X R ₁ Y	e	c	pc	ci	pci	d	o	m
e (equal)	e	c	pc	ci	pci	d	o	m
c (contains)	c	c, pc	pc	e, c, ci, o	ci, pci, o	c, pc, d, o, m	c, pc, o	c, pc, o, m
pc (properly-contains)	pc	pc	pc	c, pc, o	e, c, pc, ci, pci, o	c, pc, d, o, m	c, pc, o	c, pc, o
ci (contained-in)	ci	e, c, ci, d, o, m	c, pc, d, o, m	ci, pci	pci	d	ci, pci, d, o, m	d, m
pci (properly-contained-in)	pci	ci, pci, d, o, m	ALL	pci	pci	d	ci, pci, d, o, m	d
d (disjoint)	d	d	d	ci, pci, d, o, m	ci, pci, d, o, m	ALL	ci, pci, d, o, m	ci, pci, d, o, m
o (overlap)	o	c, pc, d, o, m	c, pc, d, o, m	ci, pci, o	ci, pci, o	c, pc, d, o, m	ALL	c, pc, d, o, m
m (meets)	m	d, m	d	ci, pci, o, m	ci, pci, o	c, pc, d, o, m	ci, pci, d, o, m	e, c, ci, d, o, m

이 되는 위상 관계의 쌍이 된다. 본 논문에서는 앞으로 이들 여덟 가지 위상 관계를 각각 e, d, o, m, c, ci, pc, pci 로 간략히 표현한다.

R1, R2, R3를 각각 객체 X와 Y, Y와 Z, Z와 X간의 위상 관계라 하자. 임의의 객체 X, Y, Z에 대하여 다음과 같은 위상 관계가 성립한다.

$$(X R1 Y) \wedge (Y R2 Z) \rightarrow (X R3 Z)$$

여기서 R1과 R2는 각각 모든 위상 관계를 가지는 집합 ALL(={e, c, pc, ci, pci, d, o, m})에서 취한 하나의 위상 관계이며, R3는 ALL의 부분 집합이 된다. 표 1은 주어진 R1과 R2를 이용하여 R3를 구하기 위한 이행 규칙 테이블(transitive rule table: TR table)[5]을 나타낸 것이다. 이행 규칙 테이블에서 R1과 R2는 각각 행과 열을 위한 인덱스 역할을 하며, 이 결과로 나타나는 엔트리 TR(R1, R2)가 R3를 의미한다. 즉, 객체 X, Y, Z가 주어지고, X, Y간의 위상 관계와 Y, Z간의 위상 관계가 주어지면, 이행 규칙 테이블은 X와 Z간의 기존에 주어지지 않았던 새로운 위상 관계 집합을 제시하는 역할을 한다.

이행 규칙 테이블내의 엔트리에서 나타나는 둘 이상의 요소들은 기존에 주어진 위상 관계들에 대하여 추리 규칙을 적용한 결과 발생 가능한 위상 관계들을 의미한다. 예를 들어, 객체 X와 Y간의 위상 관계 R1이 pci이고, 객체 Y와 Z간의 위상 관계 R2가 pc인 경우, X와 Z간의 위상 관계 R3는 e, c, pc, ci, pci, d, o, m 중의 하나로 나타남을 추론할 수 있다. 따라서 TR(pci, pc)는 {e, c, pc, ci, pci, d, o, m}가 되며, 이것은 사실상 가능한 모든 위상 관계에 해당되므로 표 1에서는 ALL로 표기하였다.

참고 문헌 [5]에서는 이항 위상 관계(binary topological relationship)에 대한 추리 규칙 테이블을 생성하는 정형화된 기법을 제안하였다. 그러나 추리 규칙 테이블은 이러한 기법을 이용하지 않고도 벤다이어그램을 이용하여 쉽게 생성할 수 있다. 이와 같이 추리 규칙 테이블의 생성 문제에 대한 많은 해결책이 이미 기존에 제시되어 있으므로 본 논문에서는 더 이상 이 문제에 대하여 자세히 언급하지 않는다.

III. 위상 관계의 추론 기법

본 장에서는 주어진 객체간의 위상 관계들을 대상으로 일관성을 검증하고, 이를 기반으로 새로운 위상 관

계를 추론하는 새로운 알고리즘을 제시한다. 제 3.1절에서는 알고리즘 기술을 위한 기본 개념을 설명하고, 제 3.2절에서는 알고리즘을 제안한다. 제 3.3절에서는 제안된 알고리즘의 동작 이해를 위한 예제를 제시한다.

1. 기본 개념

제안된 알고리즘에서는 관계 단언 행렬(relationship assertion table: RA matrix)과 이행 규칙 테이블(transitive rule table: TR table)을 입력으로 사용한다. RA 행렬은 초기에 주어지는 객체들간의 위상 관계를 n×n의 행렬 형태로 표현한다. 여기서 n은 대상이 되는 객체들의 수를 의미한다. 제안된 알고리즘의 궁극적인 목적은 RA 행렬상의 모든 가능한 두 객체 쌍 i, j에 대하여 이들간의 위상 관계를 나타내는 행렬 엔트리 RA(i,j)를 채워 넣는 것이다. 엔트리 RA(i,j)가 갖는 값은 모든 가능한 위상 관계 집합 ALL (= {e, c, pc, ci, pci, d, o, m})의 부분 집합이 된다. 만일, 두 객체간의 위상 관계가 아직 주어지지 않은 경우에도 이들간의 관계는 ALL 중의 하나인 것은 틀림이 없으므로 RA 행렬 상에서 ALL로 표기한다.

예를 들어, 여섯 개의 객체 Si(1≤i≤6)와 그들간의 위상 관계가 다음과 같이 단언되었다고 가정하자.: (S1 disjoint S2), (S1 meets S3), (S1 meets S6), (S2 overlaps S3), (S6 properly contained-in S5), (S5 properly contained-in S4), (S3 disjoint S4). 그림 2는 이러한 객체간의 위상 관계들을 RA 행렬의 형태로 표현한 것이며, 그림 3은 이를 다시 그래프 형

Node	1	2	3	4	5	6
1	e	d	m	u	u	m
2	d	e	o	u	u	u
3	m	o	e	d	u	u
4	u	u	d	e	pc	pc
5	u	u	u	pci	e	pc
6	m	u	u	u	pci	e

그림 2. RA 행렬로 표현한 객체간의 위상 관계
Fig. 2. Relationship Assertion (RA) Matrix.

태로 표기한 것이다. 그래프 형태의 표현에서 각 노드(node)는 객체를 나타내고, 노드간의 간선(edge)은 객체간의 위상 관계를 의미한다. 그림 2에서 굵은 선으로 표현된 간선은 미리 단언된 위상 관계를 나타내

고, 가는 선으로 표현된 간선은 이들로부터 추론해야 하는 새로운 위상 관계를 의미한다¹⁾.

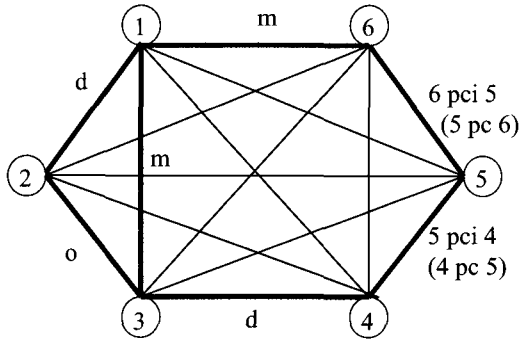


그림 3. 그래프 형태로 표현한 객체간 위상 관계
Fig. 3. Graph Representation of Topological Relationships among Region Objects.

가는 선 간선은 공통의 중간 노드를 가지며 가는 선 간선과 삼각형을 형성하는 인접한 두개의 간선을 이용하여 추론된다. 이러한 두 간선을 추리적 경로 (transitive path)라 정의한다. 따라서 하나의 새로운 간선을 유도하기 위해서는 이 간선에 대한 모든 이행 경로를 조사해야 한다. 새로운 간선을 위한 이행 경로는 그 간선의 양 끝 노드를 제외한 나머지 모든 노드들을 각각 자신의 중간 노드로 간주하므로 일반적으로 n 개의 노드들이 존재할 때 각 간선을 위하여 $n-2$ 개의 이행 경로가 존재한다. 예를 들면, 10개의 노드가 존재할 때, $45(=10C2)$ 개의 간선이 존재하게 되며, 각 간선의 추론을 위하여 $8(=10-2)$ 가지의 이행 규칙을 고려하여야 한다.

2. 알고리즘

알고리즘 1은 주어진 RA 행렬의 일관성을 검증하고, 이를 이용하여 새로운 위상 관계를 유도하기 위한 것이다. 알고리즘의 첫 부분은 이행 경로들을 형성함으로써 단언된 간선들에 대한 일관성을 검증하는 일을 수행한다. TR 테이블내의 규칙으로부터 얻어지는 위상 관계가 RA 행렬 상에 나타난 것을 포함하지 않으면, 일관되지 않음을 의미한다. 단언되지 않은 각 간선에 대해서는 대응되는 이행 경로들을 찾고, TR 테이블

1) 그림 2에서 나타난 바와 같이 두 객체간의 위상 관계는 방향성을 갖는다. 위상 관계 d, m, o, e는 대칭적(symmetric)이며, c, ci, pc, pci는 비대칭적이다. 즉, c와 ci, 그리고 pc와 pci는 각각 같은 위상 관계를 나타내지만 방향성이 다름을 의미한다. 따라서 하나의 간선은 두 객체의 입장에서 서로 다르게 해석될 수 있다.

블을 통하여 각 이행 경로에 의한 위상 관계 집합을 구한다. 또한, 이들의 교집합을 구한다. 만일, 단언되지 않은 간선에 대한 유도된 위상 관계들의 교집합이 공집합인 경우에는 이것은 주어진 위상 관계 내에 모순이 존재함을 의미한다²⁾. 교집합 내에 단 하나의 위상 관계만이 존재하는 경우에는 이를 RA 행렬의 해당 엔트리 내에 삽입한다. 이러한 정보는 이후의 처리 작업에서 다시 사용된다. 이와 같은 작업은 유도될 수 있는 더 이상의 위상 관계가 없거나, 모순이 발견될 때까지 계속 반복된다. 마지막 단계에서는 새로운 관계가 유도된 간선들을 출력하게 된다.

3. 예제

본 절에서는 제안된 알고리즘에 대한 이해를 돕기 위하여 그림 2에서 사용된 예제를 사용하여 알고리즘의 동작을 설명하고자 한다.

그림 3에 나타난 단언된 간선들은 1-2, 2-3, 3-4, 4-5, 5-6, 1-3, 1-6 등이다. 먼저, 초기에 단언된 간선들이 나타내는 위상 관계간의 일관성을 검증할 필요가 있다. 이 단언된 간선들로만 구성된 이행 경로를 갖는 간선은 1-3 하나이다. TR 테이블 내에서 간선 1-2와 2-3에 의한 결과는 {ci, pci, d, o, m}이므로 간선 1-3에 주어진 m을 포함한다. 따라서 단언된 간선들은 일관됨을 알 수 있다.

다음은 단언되지 않은 간선들을 대상으로 새로운 위상 관계를 유도해야 한다. 이 예제에서 단언되지 않은 간선들의 리스트는 1-4, 1-5, 2-4, 2-5, 2-6, 3-5, 3-6, 4-6이다. 이들 중, 간선 1-4에 대한 위상 관계는 1-2-4, 1-3-4, 1-5-4, 1-6-4의 네 가지 가능한 이행 경로를 이용하여 구할 수 있다. 그림 3에서 각 이행 경로에 속하는 간선들 중 간선 1-2, 1-3, 3-4, 5-4, 1-6에 대한 위상 관계는 각각 d, m, d, pci, m으로 단언되어 있다. 또한, 간선 2-4, 1-5, 6-4는 아직 위상 관계가 파악되지 않은 상태이므로 ALL로 표시된다. 이행 경로의 간선들과 연관된 모든 위상 관계가 결정되면, 해당 이행 경로와 대응되는 새로운 위상 관계를 유도하기 위하여 표 1에 나타난 TR 테이블을 사용한다. 즉, 간선 x-y와 y-z에 대한 라벨이 II과 I2라 할 때, TR 테이블을 적용함으로써 간선 x-z에 대한 라벨 I3을 추론된 위상 관계로서 유도한다.

2) 주어진 위상 관계간의 상호 모순은 이 알고리즘에 의해서 발견된다. 이것은 이행 규칙에 의해서 유도된 새로운 위상 관계가 주어진 것과 다른 경우에 발생한다.

```

Algorithm 1:
Input: RA matrix and TR table
Output: consistency checking and new relationships derived from RA matrix
begin
  AssertedEdges := all the edges not represented as ALL in RA matrix;
  NotAssertedEdges := all the edges represented as ALL in RA matrix;
  DerivedEdges =  $\emptyset$ ;

  /* check consistency for asserted edges by building transitive paths */
  T := all transitive paths of all the edges in AssertedEdges;
  for each transitive path  $t \in T$  do
    if assertion in RA matrix is not an element of transitive relationship in TR table
    then inconsistent_state;

  /* check consistency for not-asserted edges and derive
  new relationships from them using TR table */
  repeat
    Temp := NotAssertedEdges;
    for each not-asserted edge  $e_i \in Temp$  do begin
       $T_i$  := all transitive paths of  $e_i$ ;
       $R_i$  := ALL;
      for each transitive path  $t \in T_i$  do
        if both edges are not represented as ALL in RA matrix;
        then begin
           $r$  := transitive relationship in TR table;
           $R_i := R_i \cap r$ ;
        end
        if  $R_i = \emptyset$  then
          inconsistent_state;
        else if  $R_i$  contains only one element then begin
          insert the element into the corresponding entry in RA matrix;
          DerivedEdges := DerivedEdges  $\cup$   $e_i$ ;
          NotAssertedEdges := Temp -  $e_i$ ;
        end
      end
    end
  until Temp  $\neq$  NotAssertedEdges and NotAssertedEdges  $\neq \emptyset$ ;

  output all the edges in DerivedEdges;
end

```

예를 들면, 간선 1-3과 3-4에 대한 위상 관계가 각각 m 과 d 인 이행 경로 1-3-4에 대하여 간선 1-4와 대응되는 위상 관계는 표 1에서 나타나는 TR 테이블에 의하여 $\{c, pc, d, o, m\}$ 이 된다. 이행 경로 1-2-4, 1-5-4, 1-6-4에서와 같이 이행 경로상의 간선들 중 적어도 하나의 위상 관계를 파악할 수 없는 ALL의 경우에는 이행 경로의 위상 관계도 역시 파악할 수 없으므로 그대로 ALL이 된다. 한 간선을 위한 이행 경로들의 모든 위상 관계가 파악되면, 이들의 교집합이 그 간선의 위상 관계를 최종적으로 결정하게 된다. 본 예에서는 네 가지 집합 ALL, $\{c, pc, d, o, m\}$, ALL, ALL의 교집합이 되므로 $\{c, pc, d, o, m\}$ 이 된다. 알고리즘 1의 Repeat 문 내의 바깥쪽 for 문의 첫 순환을 통하여 얻어진 유도 결과는 다음과 같다. 즉, 첫 번째 순환의 결과, 간선 3-5, 3-6, 4-6과 관련된 위상 관계가 각각 $\{d\}$, $\{d\}$, $\{pc\}$ 로 유일하게 결정된다. 이들은 RA 행렬의 해당 엔트리 내에 삽입

되며, 다음 순환에서 아직 위상 관계가 확정되지 않은 간선의 위상 관계를 유도하는데 사용된다.

- 1) 간선 1-4

이행 경로	가능한 위상 관계
1 (d)→ 2 (ALL)→ 4	ALL
1 (m)→ 3 (d)→ 4	$\{c, pc, d, o, m\}$
1 (ALL)→ 5 (pci)→ 4	ALL
1 (m)→ 6 (ALL)→ 4	ALL

 가능한 위상 관계들의 교집합 = $\{c, pc, d, o, m\}$
- 2) 간선 3-5

이행 경로	가능한 위상 관계
3 (m)→ 1 (ALL)→ 5	ALL
3 (o)→ 2 (ALL)→ 5	ALL
3 (d)→ 4 (pc)→ 5	$\{d\}$
3 (ALL)→ 6 (pci)→ 5	ALL

 가능한 위상 관계들의 교집합 = $\{d\}$

3) 간선 3-6

이행 경로 가능한 위상 관계
 3 (m)→ 1 (m)→ 6 {e, c, ci, d, o, m}
 3 (o)→ 2 (ALL)→ 6 ALL
 3 (d)→ 4 (ALL)→ 6 ALL
 3 (d)→ 5 (pc)→ 6 {d}
 가능한 위상 관계들의 교집합 = {d}

4) 간선 4-6

이행 경로 가능한 위상 관계
 4 (ALL)→ 1 (m)→ 6 ALL
 4 (ALL)→ 2 (ALL)→ 6 ALL
 4 (d)→ 3 (d)→ 6 ALL
 4 (pc)→ 5 (pc)→ 6 {pc}
 가능한 위상 관계들의 교집합 = {pc}

두 번째 순환은 첫 순환과 마찬가지로 위상 관계가 아직 확정되지 않은 간선 1-4, 1-5, 2-4, 2-5, 2-6을 대상으로 진행된다. 이 결과, 간선 1-4, 1-5와 대응되는 위상 관계는 아래와 같은 방식을 통하여 유일한 값 {o}를 갖게 된다. 따라서 두 번째 순환에서 간선 1-4와 1-5와 대응되는 위상 관계는 유일하게 결정되지만, 간선 2-4, 2-5, 2-6과 대응되는 위상 관계는 아직 완전히 결정되지는 않는다. 간선 1-4, 1-5와 관련된 위상 관계 {o}는 RA 행렬의 해당 엔트리 내에 삽입된다.

5) 간선 1-4

이행 경로 가능한 위상 관계
 1 (d)→ 2 (ALL)→ 4 ALL
 1 (m)→ 3 (d)→ 4 {c, pc, d, o, m}
 1 (ALL)→ 5 (pci)→ 4 ALL
 1 (m)→ 6 (pci)→ 4 {ci, pci, o}
 가능한 위상 관계들의 교집합 = {o}

6) 간선 1-5

이행 경로 가능한 위상 관계
 1 (d)→ 2 (ALL)→ 5 ALL
 1 (m)→ 3 (d)→ 5 {c, pc, d, o, m}
 1 (o)→ 4 (pc)→ 5 {c, pc, d, o, m}
 1 (m)→ 6 (pci)→ 5 {ci, pci, o}
 가능한 위상 관계들의 교집합 = {o}

간선 1-4에 대한 위상 관계 = {o}

간선 1-5에 대한 위상 관계 = {o}

간선 2-4에 대한 위상 관계 = {d, o, m}

간선 2-5에 대한 위상 관계 = {d, o, m}

간선 2-6에 대한 위상 관계 = {d, o, m}

세 번째 순환은 남은 간선 2-4, 2-5, 2-6을 대상으로 진행된다. 그러나 세 번째 순환에서는 유일한 값을 가지는 위상 관계가 전혀 나타나지 않으므로 알고리즘은 여기서 중단된다.

결과적으로 제안된 알고리즘을 이용하여 그림 3에 주어진 위상 관계들을 대상으로 하여 유도한 새로운 위상 관계는(S1 overlap S4), (S1 overlap S5), (S3 disjoint S5), (S3 disjoint S6), (S4 properly-contains S6), (S2 disjoint or overlap or meet S4), (S2 disjoint or overlap or meet S5), (S2 disjoint or overlap or meet S6)이다. 첫 번째 나타나는 다섯 가지는 확정된 위상 관계를 나타내며, 이후의 세 가지는 각각 두 객체간에 나타날 수 있는 가능한 세 가지 위상 관계들을 나타낸 것이다. 그림 4는 주어진 위상 관계들과 유도된 위상 관계들을 모두 포함하여 RA 행렬 형태로 나타낸 것이다.

Node	1	2	3	4	5	6
1	e	d	m	o	o	m
2	d	e	o	d,o,m	d,o,m	d,o,m
3	m	o	e	d	d	d
4	o	d,o,m	d	e	pc	pc
5	u	d,o,m	d	pci	e	pc
6	m	d,o,m	d	pci	pci	e

그림 4. 알고리즘 1을 적용하여 새롭게 구한 RA 행렬
 Fig. 4. Relationship assertion(RA) matrix after applying algorithm 1.

제안된 알고리즘의 공헌을 요약하면 다음과 같다. 먼저 주어진 객체들간의 위상 관계들을 대상으로 상호 모순이 존재하는가를 검증하고, 이들과 이행 규칙을 제공하는 TR 테이블을 이용하여 새로운 위상 관계들을 유도해 낸다.

IV. 구 현

제안된 알고리즘은 논리 언어(logic language)로서 가장 널리 사용되고 있는 프롤로그(Prolog)로 구현되

었다. 프롤로그에서는 사실(facts)을 규칙(rules)과 구분한다. 입력으로 위상 관계와 이행 규칙 테이블은 사실(facts)로 받아들이고 새로운 관계는 규칙(rules)을 사용하여 유도하게 된다. 프롤로그 프로그래밍 언어는 추론 엔진이 언어 자체에서 제공되므로 본 논문에서와 같이 주어진 위상 관계로부터 새로운 위상 관계를 유도할 때 별도의 노력 없이 사실과 규칙을 사용하여 쉽게 구현할 수 있다. 본 장에서는 제안된 알고리즘의 구현 이슈에 대하여 논의한다.

본 알고리즘에서 사용되는 객체와 위상 관계, 그리고 TR 테이블이 사실에 해당된다. 객체, 위상 관계, TR 테이블은 프롤로그에서 다음과 같이 표현된다.

(a) 객체:

```
spatial_objects([s1, s2, s3, s4, s5, s6]).
```

여기서 spatial_objects라는 술어(predicate)는 객체들의 이름을 의미한다.

(b) 위상 관계:

```
relationship(s1, s2, o).
```

```
relationship(s1, s2, m).
```

```
relationship(s1, s6, m).
```

```
relationship(s2, s3, o).
```

```
relationship(s6, s5, pci).
```

```
relationship(s5, s4, pci).
```

```
relationship(s3, s4, d).
```

여기서 relationship이란 술어는 대상이 되는 두 객체와 그들간의 위상 관계를 인자로 취한다. 예를 들면, relationship(s1, s2, o)는 객체 s1과 s2간에 overlap이란 관계가 존재함을 의미한다.

(c) TR 테이블:

```
transitive_rule(R1, R2, R3)
```

transitive_rule이란 술어는 이행 규칙 $(X \ R1 \ Y) \wedge (Y \ R2 \ Z) \rightarrow (X \ R3 \ Z)$ 를 표현한 것이다. 예를 들어, transitive_rule(c, o, [c, pc, o])는 임의의 세 객체 X, Y, Z에 대하여 X와 Y, Y와 Z간의 위상 관계는 각각 contain과 overlap일 때, X와 Z간의 위상 관계가 {contain, properly-contained, overlap} 중의 하나가 됨을 추론할 수 있음을 의미한다. 따라서 표 1내의 모든 엔트리들은 이와 같은 형태로 표기할 수 있다. 예를 들어, TR 테이블의 두 번째 행은 다음과 같이 표현된다.

```
transitive_rule(c, e, [c]).
```

```
transitive_rule(c, c, [c, pc]).
```

```
transitive_rule(c, pc, [pc]).
```

```
transitive_rule(c, ci, [e, c, ci, o]).
```

```
transitive_rule(c, pci, [ci, pci, o]).
```

```
transitive_rule(c, d, [c, pc, d, o, m]).
```

```
transitive_rule(c, o, [c, pc, o]).
```

```
transitive_rule(c, m, [c, pc, o, m]).
```

이와 같은 transitive_rule 술어를 이용하여 이행 경로에 의한 위상 관계를 다음과 같은 Horn 절 (clause)로 표현할 수 있다.

```
relationship_by_tran_path(X, Z, R3) :- relationship(X, Y, R1), relationship(Y, Z, R2), transitive_rule(R1, R2, R3)
```

위의 (a)와 (b)는 알고리즘의 입력을 형성하며, 시스템에 의하여 입력 데이터에 따라 동적으로 단언된다. 반면, (c)는 새로운 위상 관계를 유도하기 위하여 사용되는 정적 추론 규칙들이며, 입력 데이터에 관계없이 한번 단언된다.

이러한 입력을 대상으로 본 알고리즘은 프롤로그로 구현되었다. 프롤로그의 최상위의 절들은 다음과 같다. 먼저, 객체들과 위상 관계들을 읽어들이고, 위상 관계들에 대한 일관성을 검증한다. 만일 주어진 모든 위상 관계들간에 모순이 없으면, 이들을 기반으로 새로운 위상 관계들을 유도한다. 그렇지 않으면, 이러한 수행은 종료되고, 모순이 존재함을 반환해준다. 프롤로그에서 기본적으로 제공하는 asserta 술어는 새로운 위상 관계를 유도하는 동안 동적으로 변화하는 위상 관계들을 수용하기 위하여 사용된다. 끝으로 프롤로그에서 기본적으로 제공하는 retract 술어는 헤드가 주어진 위상 관계와 일치하는 모든 절들을 동적 데이터베이스 내에서 제거하기 위하여 사용된다.

```
top :-
```

```
read_data,
```

```
find_pairs_asserted(L),
```

```
find_pairs_not_asserted(NL),
```

```
check_cons_of_ass_rel(L),
```

```
derive_new_rel_from_ass_rel(NL),
```

```
retract_all(pairs(_)),
```

```
retract_all(relationship(_, _, _)).
```

```
check_cons_of_ass_rel([]).
```



```

check_cons_of_ass_rel([Pair | L]) :-
    Pair = [X, Y],
    relationship(X, Y, R),
    get_tran_path_from_pair(Pair, TranPath),
    check_rel_from_tran_path(R, TranPath),
    check_cons_of_ass_rel(L).

check_rel_from_tran_path(R, []).
check_rel_from_tran_path(R, [TranPath | L]) :-
    TranPath = [X, Y, Z], !,
    relationship_by_tran_path(X, Y, Z, R1),
    (member(R, R1)
    ;
    writeln('consistency error')),
    check_rel_from_tran_path(R1, L).
check_rel_from_tran_path(R, [TranPath | L]) :-
    check_rel_from_tran_path(R, L).

derive_new_rel_from_ass_rel(L) :-
    asserta(pairs(L)),
    derive_new_relationship(L),
    pairs(NL),
    ((L = NL), !
    ;
    derive_new_rel_from_ass_rel(NL)).

derive_new_relationship([]).
derive_new_relationship([Pair | L]) :-
    get_tran_path_from_pair(Pair, TranPath),
    intersection_of_possible_rel(TranPath, R),
    Pair = [X, Y],
    writeln(relationship(X, Y, R)),
    (length(R, 1), !, R = [SingleRel],
    asserta(relationship(X, Y, SingleRel)),
    pairs(PL),
    delete(Pair, PL, NL),
    asserta(pairs(NL)),
    derive_new_relationship(L)
    ;
    derive_new_relationship(L)).

```

V. 결 론

본 논문에서는 부분적으로 주어진 위상 관계들의

일관성 여부를 판별하고, 이들로부터 가능한 모든 새로운 위상 관계를 유도하는 알고리즘을 제안하였다. 또한, 이를 프롤로그 언어를 이용하여 구현하는 이슈에 대하여 논의하였다. 프롤로그를 사용한 이유는 객체, 위상 관계, TR 테이블등을 사실로 단언하고, 이들을 대상으로 규칙을 적용하여 새로운 위상 관계를 유도할 수 있기 때문이다.

본 논문에서는 위상 관계를 주 대상으로 논의를 전개하였으나, 방향 관계(directional relationship)도 유사한 방법을 이용하여 새로운 관계를 유도할 수 있다. 예를 들면, 객체 X가 Y의 위에 있고, Y가 Z의 위에 있는 경우, 여기서 X가 Z의 위에 있음을 쉽게 추론할 수 있다. 또한, 위상 관계와 방향 관계가 함께 주어진 상황에서도 이들을 이용하여 새로운 관계를 추론할 수 있다. 향후 연구에서는 이와 같이 두 가지 종류의 공간 관계를 통합된 형태로 다루는 공간 관계 추론 시스템을 구성하는 방안에 대하여 연구하고자 한다.

참 고 문 헌

- [1] Beckman, N., H-P Kriegel, R. Schneider, and B. Seeger, "The R*-Trees: An Efficient and Robust Access Method for Points and Rectangles" in *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pp. 322-331, Atlantic City, New Jersey, May 1990.
- [2] Bentley, J. "Multidimensional Binary Search Trees Used for Associative Searching." *Comm. of ACM*, Vol. 18, No. 9, 1975.
- [3] Brinkhoff, T., H-P Kriegel, and B. Seeger, "Efficient Processing of Spatial Joins Using R-Trees," in *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pp. 237-246, May 1993.
- [4] Clementini, E., Di Felice, P., and van Oosterom, P., "A Small Set of Formal Topological Relationships Suitable for End-User Interaction," in *Proc. of the Third Symposium on Large Spatial Databases, SSD'93*, D. Abel and B.C. Ooi, Eds., pp. 277-295, Singapore, June 1993.
- [5] Egenhofer, M., "A Formal Definition of Binary Topological Relationships," in *Proc. of the Second Symposium on Large Spatial*

- Databases, SSD'91*, O. Gunther and H.J. Schek, Eds., pp. 143-160, Zurich, Switzerland, August 1991.
- [6] Gunther, O., and A. Buchmann, "Research Issues in Spatial Databases," *ACM SIGMOD Record*, Vol. 19, No. 4, pp. 61-68, 1990.
- [7] Gunther, O., "Efficient Computation of Spatial Joins," in *Proc. of 9th IEEE Intl. Conf. on Data Engineering*, pp. 47-57, Vienna, Austria, 1993.
- [8] Gutting, R. H., "An Introduction to Spatial Database Systems," *The VLDB Journal*, Vol. 3, No. 4, pp. 357-400, October 1994.
- [9] Guttman, A., "R-Trees: A Dynamic Index Structure for Spatial Searching," in *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pp. 47-57, Boston, Mass., June 1984.
- [10] Lo, M.-L., and C. V. Ravishankar, "Spatial Joins Using Seeded Trees," in *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pp. 209-220, Minneapolis, Minnesota, May 1994.
- [11] Nievergelt, J., H. Hinterberger, and K. Sevcik., "The Grid File: An Adaptable, Symmetric Multikey File Structure," *ACM Trans. on Database Systems*, Vol. 9, No. 1., March 1984.
- [12] Papadias, D., and T. Sellis, "Qualitative Representation of Spatial Knowledge in Two-Dimensional Space," *The VLDB Journal*, Vol. 3, No. 4, pp. 479-516. October 1994.
- [13] Papadias, D., Y. Theodoridis, T. Sellis, and M.J. Egenhofer, "Topological Relations in the World of Minimum Bounding Rectangles: A Study with R-trees," in *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pp.92-103, San Jose, California, May 1995.
- [14] Robinson, J., "The K-D-B Tree: A Search Structure for Large Multidimensional Dynamic Indexes," in *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, Ann Arbor, Michigan, April 1981.
- [15] Samet, H., "The Quadtree and Related Hierarchical Data Structures," *ACM Computing Surveys*, Vol. 16, No. 2, June 1984.
- [16] Samet, H., *The Design and Analysis of Spatial Data Structures*, Addison-Wesley, 1990.
- [17] Sellis, T., Roussopoulos, and C. Faloutsos, "The R⁺-Tree: A Dynamic Index for Multi-Dimensional Objects," in *Proc. Intl. Conf. on Very Large Data Bases*, pp, 507-518, Brighton, England, 1987.
- [18] Sistla, A. P., Y. Clement, and R. Haddad, "Reasoning about Spatial Relationships in Picture Retrieval Systems," in *Proc. Intl. Conf. on Very Large Data Bases*, pp, 570-581, Santiago, Chile, 1994.
- [19] Whang, K. Y. and Krishnamurthy, R., "The Multilevel Grid File-A Dynamic Hierarchical Multidimensional File Structure," In *Proc. 2nd Intl. Conf. on Database Systems for Advanced Applications*, pp. 449-459, Apr. 1991.

저 자 소 개

黃煥圭(正會員) 第 35卷 C編 第 3號 參照