위상구축 트랜잭션의 처리

신명진, 장인성, 이기준
부산대학교 전자계산학과

Topology Building Transaction Processing

Myoung-Jin Shin, In-Sung Jang, Ki-Joune Li
Department of Computer Science, Pusan National University

요 약

지리정보시스템의 정보를 관리하는 공간데이터베이스 관리시스템에서는 위상정보에 대한 효율적인 저장과 관리가 필요하다 이렇게 기하학적인 정보만을 가진 객체의 자료 구조에 위상정보를 추가하거나 또는 위상적인 정보만으로 이러한 지도를 만드는 작업을 위상정보 구축 작업이라고 한다 공간데이터베이스의 위상정보를 생성하는 작업은 많은 연산과 상당한 시간을 요구하므로 일종의 장기 트랜잭션이라 할 수 있다 이런 위상정보 구축 트랜잭션은 위상적 데이터의 특성을 이용하면 기존에 제안된 방법보다 효과적으로 처리할 수 있다 본 논문에서는 위상정보 생성시의 장기 트랜잭션을 위하여 정의한 위상구축 트랜잭션의 완전성을 유지하면서 하위 트랜잭션으로 나눌 수 있도록 Plane-sweeping을 이용한 위상구축 알고리즘을 제안하였다 이 방법을 이용하면 위상구축이 수행되는 동안에도 위상이 구축되어 있는 지역에 대한 질의의 결과를 보장할 수 있다

1. 서 론

위상정보는 공간정보의 한 가지 종류로서, 지리정보시스템에서 사용되는 중요한 정보 중의 하나이다[1,2] 따라서, 지리정보시스템의 정보를 관리하는 공간데이터베이스 관리시스템에서 위상정보에 대한 효율적인 저장과 관리가 필요하다 이렇게 기하학적인 정보만을 가진 객체의 자료구조에 위상정보를 추가하거나 또는 위상적인 정보만으로 이러한 지도를 만드는 작업을 위상정보 구축 작업이라고 한다[8]

일반적인 데이터베이스에서의 수정연산은 수정되는 객체에만 영향을 주거나 다른 객체에 영향을 주는 과정이 비교적 단순하다 그러나, 위상관계는 공간객체들 사이의 복잡한 연관관계로 이루어져 있다 즉, 하나의 공간객체에 대한 수정, 삽입, 삭제연산은 여러 다른 공간객체에도 영향을 미치게 된다 이러한 위상 연산 중 위상구축은 하나의 도면에 굉장히 많은 수정, 삽입, 삭제의 연산을 가지는 일종의 장기 트랜잭션이라 할 수 있다 따라서 본 논문에서는 이러한 트랜잭션을 처리하기 위하여 정의한 위상구축 트랜잭션 완전성을 유지하면서 여러 개의 하위 트랜잭션으로 나누어 처리하는 방법을 제안하였다 본 논문에서 제안한 방법으로 위상구축 트랜잭션을 처리할 경우는 트랜잭션이 처리 중이라도 구축된 위상에 대한 질의에 대하여 올바른 결과를 보장할 수 있다

본 논문의 구성은 다음과 같다 2 장에서는 위상 구축 트랜잭션의 배경에 대하여 설명하고, 3 장에서는 위상구축 트랜잭션의 완전성에 대해서 설명한다 그리고 4 장에서는 plane-sweep 을 이용하여 위상구축 트랜잭션을 하위 트랜잭션으로 나누는 방법을 설명한다 마지막으로 5 장에서는 결론과 함께 향후 연구방향을 제시한다.

2. 배 경

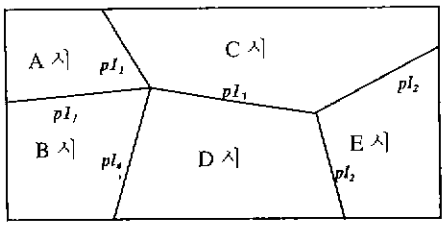
공간데이터베이스에서 위상 정보를 효율적으로 저장하고 관리하는 것은 시스템의 성능 향상을 위해 매우 중요하다 이를 위하여 공간데이터베이스에 위상 정보를 구축하는 작업이 필요하다 이 때 필요한 트랜잭션을 위상구축 트랜잭션이라고 한다 위상구축 트랜잭션의 입력은 기하학적인 정보가 되고, 결과는 기학학적인 정보에 위상적 정보가 추가된다[8]

위상구축은 하나의 도면에 대해 매우 많은 수정, 삽입, 삭제의 연산을 가지고 있는 일종의 장기 트랜잭션이다 이러한 장기 트랜잭션은 여러 개의 단기 트랜잭션인 하위 트랜잭션으로 나누어서 처리할 수 있다 이와 같은 방법의 위상구축을 위한 장기 트랜잭션처리 방법 중 가장 일반적인 것이 중첩 트랜잭션(nested transaction)이다 중첩 트랜잭션(nested transaction)이란 트랜잭션을 계층화하여 주 트랜잭션을 여러 개의 부 트랜잭션으로 구성하는 방법이다[5]

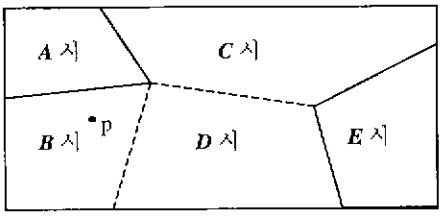
위상구축 트랜잭션을 중첩 트랜잭션(nested transaction)으로 처리할 때, 제일 단순한 방법은 각각의 도면상 기하학적 객체의 삽입 순서대로 위상을 구축하는 것이다

[그림 1]은 위상구축 할 지도의 예를 보인 것이다 이 지도는 각 A, B, C, D, E, F 의 5 개 도시를 나타내는 지도이다 그리고 시의 경계를 나타내는 다각선은 pl_i 의 i 순서대로 입력된다고 가정하자. 위의 도면을 가지고 위상구축을 하면 5 개의 면을 가지게 된다 그런데, 위상구축 트랜잭션을 도면상의 기하적 객체의 삽입 순서대로 위상을 구축하도록 중첩 트랜잭션의 하위 트랜잭션을 구성하면 [그림 2]에서와 같은 문제가 발생한다 [그림 2]는 pl_2까지 중첩 트랜잭션(nested transaction)의 하위 트랜잭션이 완료된 상태이다 로는 바와 같이 단지 A 시와 E 시의 면만 생성되었다 이때, 만일 "p 를 포함하는 도시를 찾아라"라는 질의가 주어졌다고 가정하자 그러면 결과는 B 시가 되어야 한다 그러나, 사실 B 시는 아직 위상이 구축되지 않은 상태이므로 결

의에 대한 올바른 결과를 보장 받을 수 없다



[그림 1] 위상구축 할 지도의 예



[그림 2] Pl_2 까지 위상구축이 된 상태

즉, 하위 트랜잭션에 의해서 구축된 위상이 질의의 결과를 보장하지 못하기 때문에 위상이 완전히 구축되기 전까지는 질의에 대한 올바른 결과를 보장할 수 없다는 것이다 그러나 하위 트랜잭션이 다음 절에서 설명할 위상구축 트랜잭션의 완전성을 만족하는 트랜잭션이라면 구축된 위상에 대한 질의의 올바른 결과를 보장할 수 있다

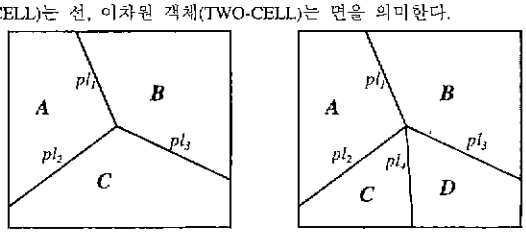
3. 위상구축 트랜잭션의 완전성 (completeness of topology building transaction)

위상구축 트랜잭션에서 위상은 트랜잭션의 진후로 일관성을 유지하고 있어야 한다 [4]에서는 위상적 일관성을 다음과 같이 정의하였다

정의 1. 위상적 일관성

- 모든 일차원 객체(ONE-CELL)는 두 개의 영차원 객체(ZERO-CELL)를 가진다
- 모든 일차원 객체(ONE-CELL)는 두 개의 이차원 객체(TWO-CELL)를 가진다
- 모든 이차원 객체(TWO-CELL)는 일차원 객체(ONE-CELL)들과 영차원 객체(ZERO-CELL)들의 하나의 사이클로 둘러 싸여 있다.
- 모든 영차원 객체(ZERO-CELL)는 일차원 객체(ONE-CELL)들과 이차원 객체(TWO-CELL)의 하나의 사이클로 둘러 싸여 있다.
- 일차원 객체(ZERO-CELL)가 없는 교차는 없다

위에서 말한 영차원 객체(ZERO-CELL)는 점을, 일차원 객체(ONE-CELL)는 선, 이차원 객체(TWO-CELL)는 면을 의미한다.



(a) pl_3가 입력되기 전 (b) pl_3가 입력된 후

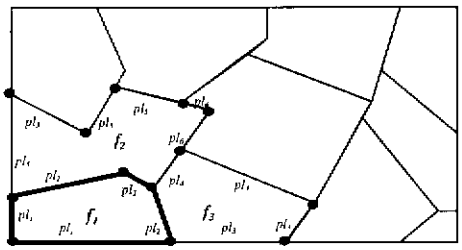
[그림 3] 기하객체 pl_3가 입력되기 전 후의 위상 질의의 결과 비교

정의 1 은 이미 구축된 위상적 데이터의 일관성을 설명하고 있을 뿐 위상구축 과정의 조건을 정의하고 있지는 않다 예를 들어, [그림 3]에서 보는 바와 같이 입력되는 기하적 객체가 $\{pl_1, pl_2, pl_3\}$일 때, 각각의 pl_i를 입력하여 현이 생성되는 위상구축 작업인 하위 트랜잭션을 T_i라고 하자 만일 (a)처럼 하위 트랜잭션이 T_3까지 처리되어 위상이 구축된 경우에도 위에서 제시한 위상적 일관성은 유지된다. 그러나, "B 지역과 인접한 지역을 찾아라"는 질의가 주어지면 정확한 답은 (b)의 경우와 같이 A 지역과 D 지역이지만 (a)의 경우 A, C 지역이 되어 그 질의의 결과를 보장할 수 없다 즉, 그와 같은 질의는 위상구축 트랜잭션이 완성되지 않았으므로 처리될 수 없다는 것이다. 이는 위상이 구축된 지역에 또 다른 기하적 객체가 삽입되는 하위 트랜잭션이 일어나기 때문이다. 이런 현상을 제거하기 위하여 위상구축 트랜잭션의 완전성(**Completeness of Topology Building Transaction**)의 상태를 아래와 같이 정의한다

정의 2. 위상구축 트랜잭션의 완전성(CTBT)

CTBT1 모든 김돌은 둘 이상의 인접한 에지를 가진다
CTBT2 모든 에지는 양끝에 각각에 서로 다른 점을 가진다.
CTBT3 모든 에지는 두개의 인접한 면을 가진다.
(에지가 도넛에 경계일 경우 하나의 인접한 면을 가진다)
CTBT4 교차하는 에지는 존재하지 않는다
CTBT5 면은 에지들로 구성된 하나의 사이클을 가진다
CTBT6 CTBT1~CTBT5 까지 만족하는 객체는 더 이상 변경되지 않는다

정의 2 와 같이 위상구축시 장기 트랜잭션을 위상구축 트랜잭션의 완전성을 만족하는 하위 트랜잭션으로 나누어서 각각을 처리하면 된다 그러나 위상구축 트랜잭션의 완전성을 만족하는 하위 트랜잭션으로 나누는 것은 불가능하다 왜냐하면, 위상이 구축되는 중간 과정의 상태는 위상구축 트랜잭션의 완전성을 만족하고 있지 않기 때문이다 결국, 전체 위상을 구축하는 작업을 하나의 트랜잭션으로 처리할 때만 가능하다 그러나 몇 개의 트랜잭션의 작업이 수행된 후에 이전의 트랜잭션이 위상구축 트랜잭션의 완전성을 만족하는 경우가 있다



[그림 4] 위상구축 트랜잭션 완전성을 만족하는 면(f_i)

[그림 4]와 같이 위상이 구축된 경우를 생각해 보자 pl_i가 i의 순서대로 입력이 되어 f_1 가 동일한 순서대로 생성되었다고 하자. f_1를 생성한 트랜잭션을 T_1라고 하면, T_1이 처리될 당시에는 T_1 은 위상구축 트랜잭션의 완전성을 만족하고 있지 않다 f_2와 f_3가 아직 구축되지 않은 상태이므로, f_1의 각 에지들은 인접한 두개의 면을 가지고 있지 않다 (CTBT3 을 위배) 그러나 T_2와 T_3를 처리한 후에는 T_2와 T_3는 역시 같은 이유로 위상구축 트랜잭션의 완전성을 만족하지 않지만, T_1은 만족하고 있다 이 때 T_1을 완료(commit)할 수 있다 이와 같은 방법으로 넓 개의 트랜잭션이 수행된 이후 그 이전 트랜잭션 중 위상구축 트랜잭션의 완전성을 만족하는 트랜잭션을 완료할 수 있다 따라서 위상구축 트랜잭션의 하위 트랜잭션을 정리하면 다음과 같다

정리 1. 위상구축 트랜잭션의 하위 트랜잭션

위상구축 트랜잭션(TBT) = $\{T_0, T_1, \ldots T_n\}$ (T_k는 하위 트랜잭션, k=0 n, $0 < i < n$)이라고 하면

- T_i는 f_i를 생성한다

- T_i 작업 후 T_k ($k < i$)가 CTBT를 만족하면 T_k를 완료(commit)한다

정리 1 과 같이 하위 트랜잭션을 나누어서 처리하면 되는데, 문제는 이러한 하위 트랜잭션을 어떤 순서대로 해야 위상구축 트랜잭션의 완전성을 만족하는 트랜잭션을 만들수 있느냐는 것이다. 최악의 경우는 마지막 하위 트랜잭션을 수행한 후 이전 모두의 트랜잭션을 원료하게 되는 경우도 생길 수 있다 따라서 하위 트랜잭션을 어떻게 순서화 하느냐가 중요한 문제가 된다 그러나 이 문제는 다음절에서 설명할 PS-PGnize 알고리즘을 이용한 위상구축 트랜잭션으로 해결할 수 있다.

4. PS-PGnize 알고리즘을 이용한 위상구축 트랜잭션

앞의 절에서 설명한 것 처럼, 위상구축 트랜잭션의 완전성을 만족하는 하위 트랜잭션의 순서를 찾기 위해서 본 논문에서는 plane-sweep 을 이용한 위상구축 트랜잭션을 제안한다 한 축에서부터 sweep-line 을 옮기면서 그 sweep-line 을 따라 순차적으로 위상을 구축하도록 하는 것이다[3]. 왼쪽에서 오른쪽으로 sweep line 을 옮기면서 sweep line이 가지는 각 기하적 객체에 대하여 에지를 생성하고, 그 에지를 포함하는 면을 생성하면서 위상을 구축한다

plane sweep 을 이용한 위상구축 알고리즘(PS-PGnize)을 구현하기 위한 자료구조는 다음과 같다

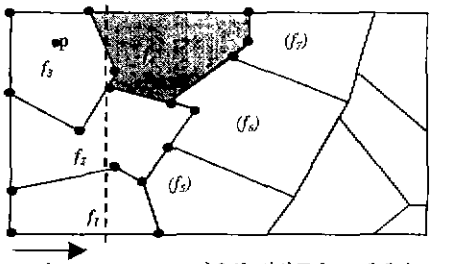
- Vertex(x coordinate, y coordinate, Edge set, oid);
- Edge(Vertex$_1$, Vertex$_2$, Face set, oid),
- Face(Edge set, oid),

Vertex 는 x 좌표와 y 좌표를 가진다 그리고 반드시 서로 다른 2 개 이상의 Edge 로 구성된 connected edge set 을 가져야 하며(CTBT1). Object identifier(oid)를 가진다. Edge 는 2 개의 Vertex (Vertex$_1$ ≠ Vertex$_2$) (CTBT2)와 두 개의 인접한 Face 를 가지고(CTBT3) Object identfier 를 가진다 Face 는 Edge set 로 구성된 하나의 사이클을 가지고(CTBT5), Object identifier(oid)를 가진다 그리고 plane sweep 을 이용하여 위상을 구축하기 때문에 교차하는 Edge 는 없다(CTBT4). 그리고, plane-sweep 을 이용하여 순차적으로 위상을 구축하므로 생성된 위상은 더 이상 수정이 되지 않는다 CTBT6).

다음은 Plane sweep 을 이용하여 위상을 구축하는 PS-PGnize 알고리즘이다

```
// 위상구축 프로시져의 시작
Procedure Topology Builder begin:
  // sweep line 과 만나는 객체로부터 Edge 하나를 생성 더 이상의 에
지가 없으면 for loop 빠져나옴
  for(i = 0 ; (Edge = sweep_line()) != NULL ; ){
    // Edge 와 인접한 Face 가 2 이하이면
    if(Edge.numoface < 2){
     //새로운 트랜잭션 시작
      Trans[i].start;
      // Edge 를 포함하고 있는 face 를 생성
      Face = make_face(Edge);
      // 생성된 Face 의 모든 edge 에 대하여
      for(Edge of Face){
        //Edge 의 다른 face 를 생성한 트랜잭션이 위상구축 트랜잭션
의 원전성(CTBT)를 만족하면
        if(Edge.otherFace.CTBT == true){
          Face_oid = Edge.otherFaceoid;
          // 해당하는 트랜잭션 완료
          Trans[Face_oid].commit;
        }
      }
      // 현재 트랜잭션이 CTBT 를 만족하면 트랜잭션 종료
      if(Face.CTBT == true){
        Trans[i].commit;
      }
      i++;
    }
  }
End Procedure: // 위상 구축 프로시저의 끝
```

[그림 5]는 위의 알고리즘에 의하여 생성된 면들이다. 먼 f_i가 i의 순서대로 생성되었다 [그림 5]에서 f_1, f_2, f_3 까지의 면이 생성되고, 트랜잭션 T_1~T_3까지 작업한 후, 트랜잭션 T_2에 의해서 f_4가 생성되면 f_3를 생성한 T_3는 위상구축 트랜잭션의 완전성을 만족한다. 이 때, T_3를 완료(commit)한다 이때, "p 를 포함하는 지역과 인접한 지역을 찾아라"와 같은 질의가 주어질 경우, 그림[5]에서 보는 바와 같이 f_2, f_4로 올바른 질의의 결과를 얻을 수 있다



[그림 5] plane sweep 이용한 위상구축 트랜잭션

그러나 위에서 설명한 바와 같이 하위 트랜잭션을 한 면의 위상이 구축되는 것을 작업 단위로 할 경우, 굉장히 많은 면을 가지고 있는 도면인 경우는 하위 트랜잭션의 수가 굉장히 많아진다 이 때, 트랜잭션의 시작과 완료에 많은 오버헤드가 생길 수 있다. 이같은 경우는 한 트랜잭션의 작업 단위를 몇 개의 면의 위상을 생성하도록 할 수 있다 [그림 5]에서 하위 트랜잭션의 작업 단위가 3 개의 면에 대한 위상을 구축하는 것이라고 가정하면, T_1 은 f_1, f_2, f_3, T2 는 f_4, f_6, f_7의 위상을 구축하는 것이 된다 T_2의 작업 후 T_1이 위상구축 트랜잭션의 완전성을 만족하므로, T_1을 완료(commit)하면 된다

5. 결 론

본 논문에서는 일종의 장기 트랜잭션인 위상구축 트랜잭션을 하위 트랜잭션으로 분할하여 처리하는 방법을 제시하였다 하위 트랜잭션이 위상적 일관성을 유지하더라도 만들어진 위상에 대해서 사용자가 질의를 할 경우 그 질의의 결과를 보장할 수 없으므로, 이를 보장할 수 있는 위상구축 트랜잭션의 원전성을 정의하였다. 위상구축 트랜잭션은 하위 트랜잭션이 위상구축 트랜잭션의 완전성을 만족하도록 순서화를 시켜야 하는데, Plane-sweep 을 이용한 위상구축 알고리즘(PL-Pgnize)으로 트랜잭션의 순서와 각 트랜잭션의 완료(commit)를 결정하였다 그리고 이 방법으로 트랜잭션을 처리할 경우는 각 트랜잭션이 위상구축 트랜잭션의 완전성을 만족하므로 위상구축 트랜잭션이 수행되는 중에도 구축된 위상에 대해서는 질의의 결과를 보장할 수 있다

본 논문에서는 위상구축 트랜잭션의 하위 트랜잭션의 순서와 트랜잭션의 완료의 시기에 대해서 고려하였다. 향후 동시적 제어를 위한 잠금(lock)에 대한 연구가 필요하다.

참고문헌

[1] R H. Guting, 'An Introduction to Spatial Database System," The VLDB Jounal, Vol 3, No 4, 1994

[2] M J Egenhofer and A U. Frank and J P. Jackson, "A Topological Data Model for Spatial Databases," Design and Implementation of Large Spatial Database Lecture Notes in Computer Science. Vol. 409, pp.271-286, Springer-Verlag, New York, NY, June 1989

[3] J Nievergelt and F. P. Preparata, "Plane-Sweep algorithms for Intersecting Geometric Figures," Proc. ACM SIGMOD 1982, pp 739-747

[4] R Laurini and D Thompson, "Fundametals of Spatial Information System," Academic press. 1992, pp.186-190

[5] J Gray, "The Transaction Concept Virtues and Limitation," proc 7th Int'l Conf. on Very Large Data Bases, pp144-154 Cannes, France. September, 1981

[6] G Groger and L Plumer,"Provably Correct and Complete Transaction Rules for GIS," Proc 5th Int'l Workshop on Advances in geographic information system, 1997, pp40-43

[7] D Kuo and V Gaede and K Taylor,"Using Constraints to Manage Long Duration Transactions in Spatial Information Systems,". To appear in the 3th IFCIS Conference on Cooperative Information Systems. 1998

[8] 민경욱, 이기준, "공간데이터베이스 관리시스템과 통합된 위상정보구축기," '97 한국정보과학회 가을 학술 발표논문집, Vol 24, No 2, pp249-252