Journal of Korea Multimedia Society Vol. 10, No. 9, September 2007(pp. 1093-1105)

MD-TIX: XML 질의의 효율적 처리를 위한 다차원 타입상속 색인기법

이 종 학†

요 약

본 논문에서는 XML 데이터베이스의 색인구조로 다차원 화일구조를 이용하는 다차원 타입상속 색인기법인 MD-TIX를 제안한다. 일차원 색인구조를 이용하는 기존의 XML 데이터베이스 색인기법에서는 타입상속 계층과 중첩요소가 포함된 복합 형태의 질의들에 대한 처리를 잘 지원하지 못한다. MD-TIX에서는 XML 데이터베이스의 중첩요소에 대한 색인기법을 위하여 이차원 타입상속 계층 색인기법(2D-THI)을 다차원으로 확장하여 사용한다. 2D-THI는 타입상속 계층의 단순요소에 대한 색인기법으로 킷값 도메인과 타입식별자 도메인으로 구성된 이차원 도메인 공간상에서 요소들의 클러스터링을 다루는 색인기법이다. 본 논문의 MD-TIX에서는 색인된 중첩요소를 표현하는 경로상의 각 타입상속 계층마다 하나의 타입식별자 도메인을 할당하여 구성된 다차원 도메인 공간상에서 색인 엔트리들의 클러스터링을 다룬다. 따라서 MD-TIX에서는 기존의 색인기법에서 지원하기 어려운 질의의 대상 범위가 타입상속 계층상의 임의의 타입들로 제한되거나, 질의에 포함된 복합요소들의 도메인이 타입상속 계층상의 임의의 타입들로 제한되는 경우에도 잘 지원할 수 있다.

MD-TIX: Multidimensional Type Inheritance Indexing for Efficient Execution of XML Queries

Jong-Hak Lee†

ABSTRACT

This paper presents a *multidimensional type inheritance indexing technique (MD-TIX)* for XML databases. We use a multidimensional file organization as the index structure. In conventional XML database indexing techniques using one-dimensional index structures, they do not efficiently handle complex queries involving both nested elements and type inheritance hierarchies. We extend a two-dimensional type hierarchy indexing technique(2D-THI) for indexing the nested elements of XML databases. 2D-THI is an indexing scheme that deals with the problem of clustering elements in a two-dimensional domain space consisting of the key value domain and the type identifier domain for indexing a simple element in a type hierarchy. In our extended scheme, we handle the clustering of the index entries in a multidimensional domain space consisting of a key value domain and multiple type identifier domains that include one type identifier domain per type hierarchy on a path expression. This scheme efficiently supports queries that involve search conditions on the nested element represented by an extended path expression. An extended path expression is a path expression in which every type hierarchy on a path can be substituted by an individual type or a subtype hierarchy.

Key words: XML Documents(XML 문서), XML Schema(XML 스키마), XML Index(XML 색인)

※ 교신저자(Corresponding Author) : 이종학, 주소 : 경북 경산시 하양읍 금락1리 330(712-702), 전화 : 053)850-2746, FAX : 053)850-2750, E-mail : jhlee11@cu.ac.kr

접수일 : 2007년 4월 12일, 완료일 : 2007년 8월 30일

† 정회원, 대구가톨릭대학교 컴퓨터정보통신공학부

1. 서 론

XML(eXtensible Markup Language)[1]는 인터넷의 급속한 발전과 더불어 대량의 정보를 효과적으로 표현하고 교환할 수 있는 새로운 데이터 표준 언어이다. 기존의 마크업 언어인 HTML (Hyper Text Markup Language)의 차세대 언어로서 사용이 간편하고 재사용성과 확장성이 뛰어나다는 장점을 가지고 있다. 이러한 장점으로 인해 XML은 전자상거래, 전자 민원 서비스, 데이터베이스, 웹 문서 작성, 웹 사이트 개발, 전자문서 교환, 무선 인터넷 콘텐츠, 전자 서명과 암호화와 같은 정보 보호 등 많은 분야에서 활용되고 있다.

XML 데이터베이스는 XML 문서를 저장하고 검색하기 위한 데이터베이스이다[2]. 이러한 XML 데이터베이스를 정의하기 위한 스키마 정의어로서 DTD(Data Type Definition)와 XML 스키마[3]가 있다. DTD는 요소(element)의 구조를 재사용할 수 없는 등 데이터 타입이 제한적으로 사용되는 단점을 가지고 있다. 따라서 타입상속(type inheritance)을 지원하는 XML 스키마가 W3C(World Wide Web Consortium)에 의해서 제안되었다. XML 스키마의 타입상속에 의해 정의된 XML 데이터베이스는 각 타입이 여러 개의 서브타입을 가질 수 있으므로 하나의 타입상속 계층을 형성한다. 따라서 XML 질의어는 이러한 데이터 모델상의 특징을 감안하여 질의의 대상을 하나의 타입 또는 특정 타입을 루트로 하는 타입상속 계층으로 지정할 수 있다.

XML로 작성된 문서를 효율적으로 저장하고 검색하기 위하여 XML 문서에 대한 질의 언어와 질의처리 등에 대한 분야가 현재 활발히 연구되고 있다. 특히 그 중 질의처리의 처리비용을 줄이기 위한 데이터베이스의 접근방법과 질의처리 최적화 기법에 관한 연구가 중요한 연구과제로 되고 있다. 데이터베이스의 색인구조는 탐색 조건에 따라 레코드들을 빠르고 효율적으로 검색하기 위하여 사용하는 데이터베이스의 접근 구조(access structure)이다[4]. 최근에 제안된 XML 데이터베이스의 중첩요소(nested element)[5]에 대한 색인기법은 XML 질의처리의 최적화에 크게 기여하는 것으로 보고되고 있으나[2], 이들 색인은 XML 데이터베이스가 가지는 타입상속 개념에 대한 고려를 하지 못하고 있다. 즉, 지금까지 사용되고 있는 중첩요소에 관한 색인기법으로는 DataGuide[6], 1-Index[7], Index Fabric[8], APEX [9] 등이 있으며, 이들은 구조 요약(structural summary)[10]이나 경로 색인(path index)[6-9]을 이용하여 주어진 경로 표현식에 대하여 XML 데이터베이스의 관련 있는 부분만은 검색할 수 있도록 하여 XML 데이터의 검색 속도를 향상시키는 색인기법으로 타입상속에 의한 질의를 고려하지 못하고 있다.

본 논문에서는 XML 데이터베이스의 중첩요소에 대한 색인기법에서 타입상속에 의한 XML 질의의 효율적 처리를 지원하기 위하여 다차원 동적 파일구조를 색인구조로 이용하는 새로운 색인기법을 제안하고 이를 다차원 타입상속 색인기법(MD-TIX: Multidimensional Type Inheritance Indexing)이라 한다. B^+-tree[11]와 같은 일차원 색인구조에서는 클러스터링 특성이 하나의 속성에 의해서 독점되는 반면에, 다차원 동적 파일구조는 다차원 클러스터링을 지원하는 파일구조로서 클러스터링의 특성이 파일을 구성하는 여러 속성들에 의해서 공유된다[12]. 다차원 파일구조에 대한 연구는 지금까지 많이 진행되어 왔으며, 대표적인 예로는 KD-트리[13], K-D-B-트리[14], 및 hB-트리[15] 등을 비롯하여, 그리드 파일(grid file)[16], 계층 그리드 파일(multilevel grid file)[17] 등이 있다.

본 논문에서 제안하는 MD-TIX는 타입상속 계층의 단순요소(simple element)에 대한 색인기법으로 이차원 파일구조를 이용하는 이차원 타입상속 색인기법(2D-THI)[18]을 다차원으로 확장하여 중첩요소에 대한 색인기법으로 이용하는 기법이다. 2D-THI는 이차원 파일구조를 사용하여 색인된 단순요소의 킷값 도메인과 함께 타입식별자 도메인으로 구성된 이차원 도메인 공간상의 색인 엔트리들의 클러스터링 문제를 다룬다. 즉 2D-THI에서는 사용자 질의 패턴에 따른 최적의 이차원 타입상속 색인구조를 구성한다.

이차원 타입상속 색인기법에서 이용하는 이차원 파일구조를 다차원 파일구조로 확장하여 XML 데이터베이스의 중첩요소에 대한 색인기법으로 확장할 수 있다. 즉, 다차원 파일구조를 이용한 다차원 타입상속 색인기법에서는 중첩요소의 킷값 도메인과 함께 경로 표현식에 나타나는 각 복합요소마다 한 축의 타입식별자 도메인을 할당하여 구성된 다차원 도메인 공간에 주어진 색인 엔트리들의 클러스터링 문제를 다룬다. 이와 같은 색인기법에서는

기존의 B^+-tree와 같은 일차원 색인구조를 이용한 색인기법들에서 문제가 되는 질의의 대상 범위가 타입상속 계층상의 임의의 타입들로 제한되거나, 경로 표현식에 나타나는 복합요소의 도메인이 타입상속 계층상의 임의의 타입들로 제한이 되는 XML 질의들의 처리를 한 번의 색인 탐색으로 가능하다.

본 논문의 구성은 다음과 같다. 제 2절에서는 관련 연구로서 XML 데이터베이스의 색인 구축에 필요한 기본 개념들을 살펴보고, 제 3절에서는 XML 데이터베이스의 중첩요소에 대한 색인기법으로 다차원 파일구조를 이용하는 두 가지 형태의 다차원 타입상속 색인기법을 제안한다. 제 4절에서는 제안된 두 가지 형태의 색인기법에 대한 성능을 비교 평가하고 그 결과를 제시한다. 마지막으로, 제 5절에서는 결론과 향후 연구방향을 기술한다.

2. 관련 연구

본 절에서는 XML 데이터베이스의 타입상속 색인기법을 논하는데 필요한 기본 개념들을 기술한다. XML은 현재 많은 기관과 산업체에서 정보의 관리와 교환을 위하여 사용하고 있다. 또한 여러 응용분야에서의 활용을 목적으로 폭넓은 연구를 하고 있으며, XML 스키마[3], XQuery[19] 등과 같은 XML 관련 기술에 대한 표준이 제안되어 있다. 본 절에서는 먼저 이러한 표준들에 대하여 소개한 다음 XML 질의어의 특징과 기존의 XML 색인기법들에 관하여 기술한다.

XML 스키마(XML Schema)는 XML 문서의 구조를 정의하기 위하여 제안된 XML 문서 정의어이다[3]. 그림 1은 Persons 타입에 대한 XML 스키마 그래프의 예이다. 그림에서 타입은 네모로, 요소는 동그라미로 나타내며, 타입 간의 상속관계는 화살표가 있는 점선으로 나타낸다. 그리고 요소와 타입 간의 중첩관계를 화살표가 있는 실선으로 나타내며, 해당 요소와 타입은 일반 실선으로 나타낸다. 그림 1에서 Persons 타입은 서브 타입인 Employees 타입과 Students 타입, 그리고 Employees 타입과 Students 타입의 서브 타입들을 포함하는 XML 타입상속 계층구조와 복합요소인 hometown의 도메인 타입인 Regions 타입을 포함하는 XML 타입 집단화 계층구조의 루트이다.

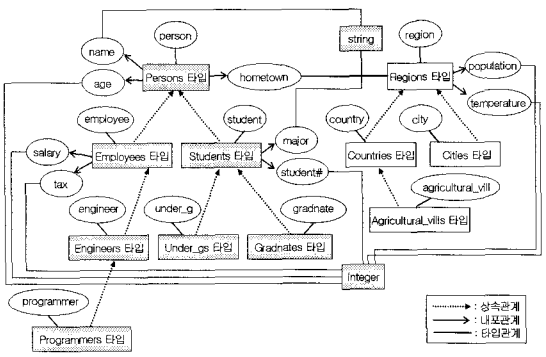


그림 1. Persons 타입에 대한 XML 스키마 그래프

타입상속 계층에서 임의의 타입 T와 그의 모든 서브 타입들을 원소로 하는 집합을 T^*로 표기한다. 예를 들어 그림 1에서 $Persons^*$ 타입은 집합 {Persons 타입, Employees 타입, Students 타입, Engineers 타입, Undergraduates 타입, Graduates 타입, Programmers 타입}이고, $Students^*$ 타입은 {Students 타입, Undergraduates 타입, Graduates 타입}이다.

XQuery는 XML 데이터베이스에서의 질의어로 FLWR(FOR, LET, WHERE, RETURN) 절로 구성이 된다[16]. FOR 절은 SQL 질의의 FROM 절과 의미상으로 유사하며, LET 절은 표현을 간략하게 하기 위해서 복잡한 식을 변수 이름에 배치할 수 있도록 한 것이다. WHERE 절은 SQL에서의 WHERE 절과 유사하며 단순 요소에 대한 조건인 단순술어(simple predicate)와 함께 중첩요소에 대한 조건인 중첩술어(nested predicate)를 사용할 수 있다. 그림 2는 그림 1에서 "인구가 50,000명 이상인 지역이 고향인 사람들의 이름을 검색하라"는 질의를 XQuery로 작성한 예문이다.

XPath(XML Path Language)는 중첩요소의 경로를 표현하기 위한 경로 표현식(path expression)이다[1]. 본 논문에서는 경로 표현식에서 경로상의 요소의 타입을 타입상속 계층상의 일부 타입들로

```
FOR $h IN Persons*
WHERE $h/hometown[population >= 50000]
RETURN <name> $h/name </name>
```

그림 2. XQuery 예문

한정하여 표현할 수 있도록 XPath를 확장하여 이를 *확장된 XPath*라 한다. 확장된 XPath는 각 요소 다음에 타입의 이름이 올 수 있도록 확장한 것으로 다음과 같은 형태를 가진다. 단, E_i 뒤의 중괄호{ }는 선택적임을 나타내는 표시이다.

$$EP = T_1/E_1\{(T_2)\}/E_2\{(T_3)\}/ \ldots /E_n\{(T_{n+1})\} \quad (1)$$

경로 *EP*에서 타입 T_1을 *타겟타입*, T_{i+1}을 요소 E_i의 *도메인타입*이라 정의한다. 타겟타입과 도메인타입은 경로에서 타입상속 계층구조에 속하는 특정 타입으로 한정(limit)될 수 있으며, 이를 타입 대치(type substitution)라 한다. 이러한 타입 대치는 질의의 범위를 특정 타입으로 한정할 수 있도록 하여 타입상속의 개념을 XML 질의에 표현하도록 한 것이다. 다음 중첩술어들은 그림 2의 질의로부터 확장된 XPath로 표현된 타입 대치에 대한 예를 보여주고 있다.

Pn1: Employees*/hometown[population >= 50000]

Pn2: Employees/hometown[population >= 50000]

Pn3: Employees*/hometown(Countries*)
[population >= 50000]

Pn4: Employees/hometown(Countries)
[population >= 50000]

중첩술어 *Pn1*은 질의 대상을 *Persons* 타입의 타입상속 계층에서 Employees* 즉, Employees 타입, Engineers 타입, Programmers 타입에 속하는 요소들로 한정하는 조건식이며, *Pn2*는 질의 대상을 Employees 타입에만 속하는 요소들로 한정하는 조건식이다. *Pn3*은 *Pn1*에서 복합요소 hometown의 타입을 Countries* 즉, Countries 타입, Agricultural-vills 타입으로 한정하는 조건식이며, *Pn4*는 *Pn2*에서 복합요소 hometown의 타입을 Countries 타입만으로 한정하는 조건식이다.

확장된 XPath식 *EP*에서 *경로 인스턴스(path instance)*는 다음 조건을 만족하는 요소들의 리스트 (E_1, E_2, ..., E_{n+1})로 정의한다. (1) 요소 E_1은 타입 T_1의 요소이다. (2) 요소 E_i ($1 < i <= n+1$)는 타입 T_i의 요소로서 요소 E_{i-1}의 구성 요소이다.

지금까지 제안된 XML 데이터베이스의 중첩요소에 대한 색인기법으로는, DataGuide[6], 1-Index [7], Index Fabric[8], APEX[9] 등이 있다. DataGuide는 비결정적(non-deterministic) 오토마타를 결정적(deterministic) 오토마타로 변환하는 과정과 동일한 과정으로 경로를 색인하는 기법이다. 일반적으로 비결정적 오토마타를 결정적 오토마타로 바꿀 경우, 크기가 커지게 되지만, XML 문서 내에 동일한 경로들이 많이 존재할수록 색인의 크기는 줄어든다. 1-index는 루트로부터 시작되는 경로의 집합이 동일한 노드들을 모아 색인을 구축하는 기법으로서, DataGuide와 마찬가지로 XML 문서 내에 동일한 경로가 매우 많이 존재한다는 점을 이용하는 색인기법이다. 그러나 이러한 기존의 색인기법들은 B^+-tree와 같은 일차원 색인구조를 이용한다. 따라서, XML 데이터 모델의 타입상속의 특징을 반영하지 못하는 것들로써, 타겟 타입의 대치 또는 도메인타입의 대치가 있는 질의는 지원하지 못한다. 즉, 질의의 대상 범위가 타입상속 계층상의 임의의 타입들로 제한되거나, XPath식에 나타나는 어떠한 요소의 도메인 타입이 타입상속 계층상의 임의의 타입들로 제한이 되는 질의들을 지원할 수 없다.

3. 중첩요소에 대한 다차원 타입상속 색인기법

본 절에서는 XML 데이터베이스의 중첩요소에 대한 색인기법으로 다차원 파일구조를 이용하는 다차원 타입상속 색인기법에 관하여 논의한다. 먼저, 제 3.1절에서는 XML 데이터베이스의 질의에 나타나는 중첩술어들의 특징을 살펴보고, 제 3.2절과 제 3.3절에서는 이러한 중첩술어를 만족하는 객체들의 신속한 탐색을 위한 색인구조들을 제안한다. 그리고 제 3.4절에서는 이들에 대한 운영 알고리즘을 제시한다.

3.1 중첩술어의 특징

그림 3은 XML 데이터베이스의 스키마 그래프에서 두 개의 타입상속 계층으로 이루어진 하나의 경로를 나타내는 경로 스키마이다. 그림 3의 경로 스키마에서, 타입상속 계층 A에서 정의된 복합요소 a의 도메인은 타입상속 계층 B이며, 타입상속 계층 B에서 정의된 단순요소 b는 타입상속 계층 A의 중첩요소가 된다. A′와 B′는 각각 타입상속 계층 A와 B의 서브타입상속 계층을 나타낸다.

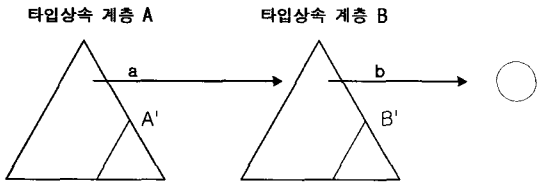


그림 3. XML 데이터베이스의 경로 스키마

그림 3과 같은 경로 스키마에 대하여 중첩요소 b에 조건이 주어진 중첩술어에는 중첩요소를 표현하는 XPath에 따라 다음과 같은 세 가지 형태로 분류할 수 있다. 여기서 Const는 중첩요소 b에 조건이 주어지는 조건 값이다.

- 중첩술어 1 : A/a/19b θ Const]
- 중첩술어 2 : A′/a/19b θ Const]
- 중첩술어 3 : A′/a(B′)/19b θ Const]

이들은 모두 중첩요소 b에 조건이 주어진 술어들이지만, 각 중첩술어를 만족하는 결과는 서로 다르게 된다. 즉, 중첩술어 1은 타입상속 계층 A의 모든 타입을 질의의 대상으로 중첩요소 b에 조건이 주어진 술어이고, 중첩술어 2는 중첩술어 1에서 질의의 대상을 타입상속 계층 A에서 A′에 속하는 타입들만으로 한정한 것이다. 그리고 중첩술어 3은 중첩술어에서 복합요소 a의 도메인으로 타입상속 계층 B에서 B′에 속하는 타입들로 한정한 것이다. 중첩술어 2에서 사용된 XPath에는 타겟 타입 대치가 있는 경우이며, 중첩술어 3에서 사용된 XPath에는 타겟 타입 대치와 도메인타입 대치가 있는 경우이다.

3.2 다차원 타입상속 색인구조

중첩요소에 대한 다차원 타입상속 색인구조에서는 킷값 도메인과 함께 중첩요소를 표현하는 경로상의 각 타입상속 계층마다 타입식별자들로 구성된 한 차원씩의 타입식별자 도메인을 할당하여 (1 + *경로 길이*) 차원의 도메인 공간을 구성한다. 예를 들어, 그림 3과 같은 경로 스키마에서 중첩요소 b에 대한 색인구조로서 삼차원 색인구조를 이용하여 다음과 같은 삼차원 도메인 공간을 구성할 수 있다. 즉, 첫 번째 축은 중첩요소 b의 킷값 도메인을 할당하고, 두 번째 축과 세 번째 축은 각각 타입상속 계층 A의 타입식별자 도메인과 타입상속 계층 B의 타입식별자 도메인을 할당한다. 그림 4는 이와 같이 구성된 삼차원 도메인 공간상에서 중첩술어 3에 대한 질의 영역을 표현한 것이다.

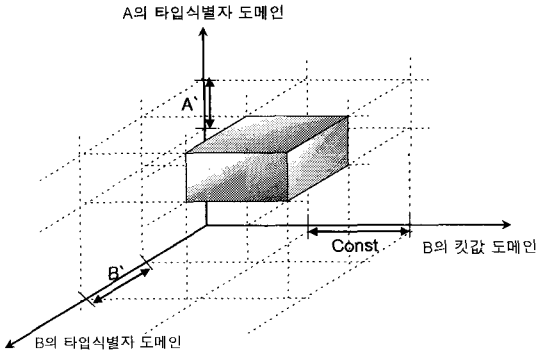


그림 4. 중첩술어 3에 대한 질의 영역

본 논문에서는 중첩요소에 대한 다차원 타입상속 색인구조를 다차원 파일구조의 하나인 계층 그리드 파일(multilevel grid file: MLGF)[17]을 이용하여 구성한다. MLGF는 디렉토리와 색인 페이지로 구성 된다[1]. 디렉토리는 다단계의 균형 트리 구조를 가지며, 색인 페이지는 최하위 단계의 디렉토리 레코드에 의해서 표현된 영역 내에 속하는 색인 레코드들을 저장한다. 색인 페이지의 색인 레코드에는 경로상의 각 타입식별자 값(type-id value) 필드, 킷값(key value) 필드, 요소 또는 경로의 개수 필드, 및 이들에 대한 색인 엔트리들의 리스트 필드가 있다. 그리고 레코드의 크기가 페이지의 크기보다 크게될 때 오버플로우 페이지를 할당하고 이를 포인트 하기 위한 오버플로우 페이지(overflow page) 필드가 있다.

본 논문에서는 다차원 타입상속 색인구조의 색인 레코드에 있는 색인 엔트리의 구성방법에 따라 다차원 중첩요소 색인구조와 다차원 경로 색인구조의 두 가지 색인구조로 분류한다. *다차원 중첩요소 색인구조*는 색인 엔트리를 색인된 중첩요소의 타겟 타입상속 계층에 속하는 요소에 대한 요소 식별자(즉, Eid)들로 구성하며, *다차원 경로 색인구조*는 색인 엔트리를 색인된 중첩요소에 대한 경로 인스턴스(즉, Eid 리스트)들로 구성한다. 그림 5는 다차원 중첩요소 색인구조의 색인 페이지 구조를 나타내며, 그림 6은 다차원 경로 색인구조의 색인 페이지 구조를 나타낸다.

1) MLGF의 디렉토리 페이지와 색인 페이지는 B-tree의 비 단말(non-leaf) 페이지와 단말(leaf) 페이지에 해당한다.

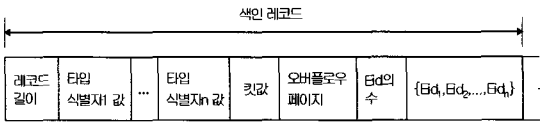


그림 5. 다차원 중첩요소 색인구조의 색인 페이지 구조

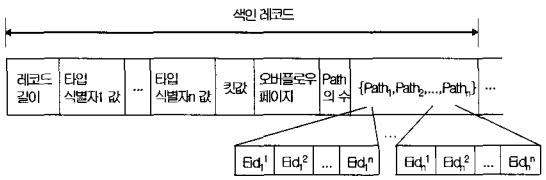


그림 6. 다차원 경로 색인구조의 색인 페이지 구조

3.3 타입식별자 도메인의 구성

다차원 타입상속 색인구조의 타입식별자 *도메인*은 타입상속 계층을 이루는 타입들의 식별자들로 구성하며, 하나의 타입상속 계층에서 임의의 서브타입 상속 계층에 포함되는 타입식별자들이 도메인상에서 하나의 구간이 되도록 구성하여야 한다. 이는 질의의 대상이 서브타입상속 계층으로 제한된 중첩 요소에 대한 질의의 처리를 위한 색인 탐색이 다차원 도메인 공간상에서 하나의 영역 탐색으로 가능하여야 하기 때문이다.

타입상속 계층에서 임의의 타입 T를 루트로 하는 서브타입상속 계층의 모든 타입 집합을 T^*로 표현할 때, 타입식별자 도메인을 타입식별자들이 타입상속 계층의 전위 순회(preorder traversal)의 순서[20]로 나열되게 구성함으로써, T^*에 포함되는 타입식별자들이 도메인 상에서 하나의 구간으로 표현되게 할 수 있다. 따라서 T^*를 질의의 대상으로 색인된 요소에 조건이 주어진 질의의 처리를 위한 색인 탐색이 다차원 도메인 공간상에서 하나의 영역 탐색으로 가능하게 할 수 있다.

그림 7은 이와 같은 원리를 이용하여 타입식별자 도메인을 구성하는 T^*-domain 알고리즘을 나타낸다. T^*-domain 알고리즘을 타입상속 계층의 루트 타입의 식별자(Type-id)와 초기치(init-value: 0)를 매개변수로 하여 수행시킴으로서 타입식별자 도메인 상에서 각 타입 집합 T^*에 대응하는 연속된 값들의 범위(init-value ~ last-value)를 구할 수 있다.

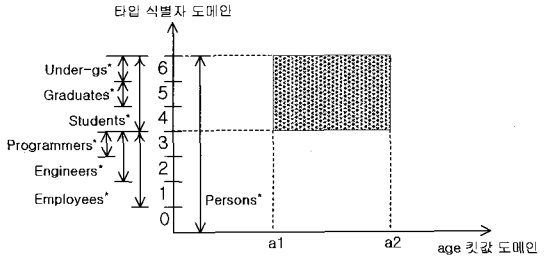
```
int T*-domain(Type-id, int init);
    {
        int last;
        init-value of Type-id* = init;
        last = init;
        for ( each child of Type-id ) {
            last = T*-domain(Type-id of child,
                last+1);
            }
        last-value of Type-id* = last;
        return last;
    }
```

그림 7. T^*-domain 알고리즘

예를 들어, 그림 8은 그림 1의 스키마 그래프에서 Persons 타입상속 계층에 대한 타입식별자 도메인의 구성 예를 나타낸다. 그림 8(a)는 각 타입식별자들이 Persons 타입상속 계층의 전위 순회 순서로 매핑되는 도메인상의 값과 각 타입 T를 루트로 하는 타입 집합 T^*에 대응하는 타입식별자 도메인상의 연속된 값들의 범위를 나타내고, 그림 8(b)는 $Students^*$에 대한 요소 age의 범위 (a1<age<a2) 질의에 대한 색인 영역을 이차원 색인 공간에서 표현한 것이다.

타입 식별자	도메인상의 T값	T^*의 범위
Persons	0	0 ~ 6
Employees	1	1 ~ 3
Engineers	2	2 ~ 3
Programmers	3	3 ~ 3
Students	4	4 ~ 6
Under-gs	5	5 ~ 5
Graduates	6	6 ~ 6

(a)



(b)

그림 8. Persons 타입상속 계층의 타입 식별자 도메인

3.4 다차원 타입상속 색인구조의 운영 알고리즘

3.4.1 다차원 중첩요소 색인구조

주어진 경로 $EP = T_1/E_1/E_2 \ldots /E_n$과 이 경로를 지원하는 다차원 중첩요소 색인구조에서는 타입상속 계층 T_1의 중첩요소 E_n에 조건이 주어진 중첩술어를 만족하는 요소들을 한번의 색인 검색으로 탐색이 가능하다. 즉, 이와 같은 색인 검색은 확장된 XPath에서 타입 대치된 타입들의 식별자 값들과 키 요소 E_n의 값으로 구성된 색인키 리스트에 의한 다중 키 검색이 된다.

다차원 중첩요소 색인구조는 경로 인스턴스를 유지하는 다차원 경로 색인구조에 비해 저장 공간의 오버헤드가 적은 반면에, 데이터베이스의 변경에 따른 색인구조의 유지 비용에 대한 오버헤드가 많다. 경로 EP에서 i번째 타입상속 계층에 있는 임의의 요소 E_i의 값으로 E_{i+1}에서 새로운 E'_{i+1}로 변경될 경우, 다차원 중첩요소 색인구조의 갱신을 위해서는 다음과 같은 단계의 작업이 필요하다.

첫 번째 단계로, 요소 E_{i+1}로부터 중첩요소 E_n까지 경로상의 타입식별자 값들과 함께 키 요소 값으로 구성된 색인키 리스트들의 집합 KS(A)를 구한다. 이를 위해서는 요소 E_{i+1}로부터 경로를 따라 순방향 운행을 하여야 한다. 여기서, 경로상의 임의의 요소 E_i가 다중 값을 갖지 않으면 KS(A)는 하나의 원소만을 가진다.

두 번째 단계로, 요소 E'_{i+1}로부터 중첩요소 E_n까지 경로상의 타입식별자 값들과 함께 키 요소 값으로 구성된 색인키 리스트들의 집합 KS(B)를 구한다. 여기서, KS(A) = KS(B)이면 색인의 갱신이 필요 없으며, 그렇지 않으면 세 번째 단계를 시행한다.

세 번째 단계로, E_i를 직접 또는 간접적으로 참조하는 타겟 타입상속 계층 T_1의 요소 Eid들의 집합 ES를 구한다. 이를 위해서는 요소 E_i로부터 경로를 따라 역방향 운행을 하여야 한다. 여기서, i = 1이면 ES는 $\{E_i\}$가 된다.

네 번째 단계로, 다음과 같이 경우에 따라 색인을 갱신한다. KS(A) ⊃ KS(B) 이면, {KS(A) - KS(B)}를 R로 하고 R에 속하는 각 색인키 리스트에 해당하는 색인 레코드에서 집합 ES에 있는 Eid들을 제거한다. 그리고 KS(A) ⊂ KS(B) 이면, {KS(B) - KS(A)}를 R로 하고 R에 속하는 각 색인키 리스트에 해당하는 색인 레코드에서 집합 ES에 있는 Eid들을 첨가한다. 두 가지 경우가 모두 아니라면, {KS(A) - KS(B)}를 R1, {KS(B) - KS(A)}를 R2라 하고, R1에 속하는 각 색인키 리스트에 해당하는 색인 레코드에서 집합 ES에 있는 Eid들을 제거하고, R2 에 속하는 각 색인키 리스트에 해당하는 색인 레코드에서 집합 ES에 있는 Eid들을 첨가한다.

이와 같이 다차원 중첩요소 색인구조에서는 데이터베이스의 변경에 따른 색인구조의 갱신을 위해서 경로상의 역방향 운행이 필요하다. 그리고 요소의 삽입과 제거에 따른 색인구조의 갱신을 위해서는 한 번의 순방향 운행만이 필요하며, 이는 변경의 경우와 같다.

3.4.2 다차원 경로 색인구조

주어진 경로 $EP = T_1/E_1/E_2 \ldots /E_n$과 이 경로를 지원하는 다차원 경로 색인구조에서는 경로상의 임의의 타입상속 계층 T_i에 대해 중첩요소 E_n에 조건이 주어진 중첩술어를 만족하는 요소들을 한번의 색인 검색으로 탐색이 가능하다. 즉, 이와 같은 색인 검색은 확장된 XPath에서 타입 대치된 타입들의 식별자 값들과 키 요소 값으로 구성된 색인키 리스트에 의한 색인 검색으로 구해진 경로 인스턴스들 중에서 i번째 항목을 프로젝션하여 구할 수 있다.

다차원 경로 색인구조는 색인 레코드에 경로 인스턴스를 저장하기 때문에 다차원 중첩요소 색인구조에 비해 많은 양의 저장 공간을 필요로 하는 반면에, 데이터베이스의 변경에 따른 색인구조의 유지 비용에 대한 오버헤드가 다차원 중첩요소 색인구조에 비해 적게 된다. 경로 EP에서 i번째 타입상속 계층에 있는 임의의 요소 E_i의 값으로 E_{i+1}에서 새로운 E'_{i+1}로 변경될 경우, 다차원 경로 색인구조의 갱신을 위해서는 다음과 같은 단계의 작업이 필요하다.

첫 번째 단계로, 요소 E_{i+1}로부터 중첩요소 E_n까지 경로상의 타입식별자 값들과 함께 키 요소 값으로 구성된 색인키 리스트들의 집합 KS(A)를 구한다. 이를 위해서는 요소 E_{i+1}로부터 경로를 따라 순방향 운행을 하여야 한다.

두 번째 단계로, 요소 E'_{i+1}로부터 경로에 따른 순방향 운행을 통하여 중첩요소 E_n까지 서브경로 인스턴스들의 집합 PI.suf와 경로상의 타입식별자 값들과 키 요소 값으로 구성된 색인키 리스트들의 집합

KS(B)를 구한다.

세 번째 단계로, KS(A)에 있는 각 색인키 리스트에 해당하는 색인 레코드를 액세스하여 i번째 항목이 E_i이고 $i+1$번째 항목이 E_{i+1}인 경로 인스턴스를 삭제함과 동시에, 각 경로 인스턴스의 첫 번째 항목에서 i번째 항목까지의 부분을 PI.pre에 보관한다.

네 번째 단계로, i번째 항목이 E_i이고 $i+1$번째 항목이 E'_{i+1}인 새로운 경로 인스턴스 집합 PI를 생성한다. 이는 PI.pre에 있는 요소들과 PI.suf에 있는 요소들을 각각 연결함으로써 얻을 수 있다.

다섯 번째 단계로, KS(B)에 있는 각 색인키 리스트에 해당하는 색인 레코드에 집합 PI에 있는 경로 인스턴스를 첨가한다.

이와 같이 다차원 경로 색인구조에서는 다차원 중첩요소 색인구조에서와는 달리 데이터베이스의 변경에 따른 색인구조의 갱신을 위한 역방향 운행이 필요 없다. 이는 경로 인스턴스가 색인 레코드에 저장되어 있기 때문이다. 따라서 다차원 경로 색인구조는 데이터베이스의 요소 내에 역 참조자가 존재하지 않아도 사용할 수 있다.

4. 성능 평가

본 논문에서 제안한 두 색인구조의 검색 성능에 대하여 살펴보면, 색인 엔트리를 색인된 중첩요소의 경로 인스턴스로 구성하는 다차원 경로 색인구조가 색인 엔트리를 타겟 타입상속 계층에 속하는 요소 식별자만으로 구성하는 다차원 중첩요소 색인구조에 비하여 저장 공간의 오버헤드로 인하여 검색 성능이 떨어지게 된다. 즉, 검색 성능은 다차원 중첩요소 색인구조가 다차원 경로 색인구조보다 좋게 된다.

그러나 제 3.4절에서 살펴본 바와 같이 각 색인구조의 운영에 따른 유지비용에 대한 오버헤드는 색인된 중첩요소의 경로 길이에 따라 많은 차이를 보이게 된다. 따라서 본 절에서는 각 색인구조의 운영에 따른 유지비용을 비교 평가함으로써, 색인구조를 유지하는 경로의 길이에 따라 적합한 색인구조를 선택할 수 있는 기준을 제시한다.

경로 $EP = T_1/E_1/E_2 \ldots /E_n$에서 i번째 타입상속 계층에 있는 임의의 요소 E_i의 값으로 E_{i+1}에서 새로운 E'_{i+1}로 변경될 경우, 경로 EP에 대해 구축된 색인구조도 변경되어야 한다. 먼저, 이러한 데이터베이스의 변경에 따른 각 색인구조의 갱신을 위한 유지비용을 모델링하고, 각 색인구조의 유지를 위한 오버헤드를 상호 비교한다.

4.1 비용 모델

다음은 비용 모델에서 사용된 매개변수들에 대한 정의이다.

P: 페이지의]크기
h: 다차원 색인구조의 디렉토리 트리의 높이
NS: 다차원 중첩요소 색인구조의 색인 페이지 크기
PS: 다차원 경로 색인구조의 색인 페이지 크기
S_i: 타입상속 계층 T_i에서 요소 E_i가 가지는 동일한 값의 평균 개수
C_i: 타입상속 계층 T_i에서 요소 E_i가 가지는 유일 값의 개수
FTC: 순방향 운행에 따른 비용
BTC: 역방향 운행에 따른 비용
IMC: 색인구조의 갱신 비용

4.1.1 다차원 중첩요소 색인구조의 유지비용

먼저, 데이터베이스의 각 요소에는 자신을 참조하는 요소에 대한 역 참조자가 존재한다고 가정한다. 제 3.3절에서 언급한 것처럼 데이터베이스의 변경에 따라 다차원 중첩요소 색인구조를 갱신하기 위해서는 데이터베이스 내의 경로를 따라 두 번의 순방향 운행과 한 번의 역방향 운행이 필요하다. 그리고 색인구조 자체의 갱신이 한번 필요하게 된다. 역방향 운행과 색인구조의 갱신은 요소 E_{i+1}에 대한 검색키 값이 요소 E'_{i+1}의 것과 다를 경우에만 필요하므로, 다차원 중첩요소 색인구조의 유지비용 MC는 다음의 식 (2)와 같다.

$$MC = (2 \times FTC) + pt \times (BTC + IMC) \quad (2)$$

여기서, pt는 요소 E_{i+1}에 대한 검색키 값이 요소 E'_{i+1}의 것과 다를 확률이다. FTC는 경로의 순방향 운행에 따른 비용이고, BTC는 경로의 역방향 운행에 따른 비용이며 IMC는 색인구조의 갱신 비용이다. 순방향 운행을 위해서 액세스해야할 요소의 수는 $(n - i)$개이고, 각 요소를 액세스하기 위한 경로의 순방향 운행에 따른 비용 FTC는 식 (3)과 같다.

$$FTC = 2 \times (n - i) \quad (3)$$

그리고 역방향 운행을 위해서 액세스해야 할 요소의 개수 No는 식 (4)와 같다.

$$No = \begin{cases} \sum_{j=2}^{i-1} \left(\Pi_{k=j}^{i-1} S_k \right) & \text{if } (i > 2) \\ 0 & otherwise \end{cases} \quad (4)$$

따라서 각 요소에 대해서 두 번의 입출력이 필요하므로, BTC는 식 (5)와 같다.

$$BTC = \begin{cases} 2 \times No & \text{if}(i > 2) \\ 0 & otherwise \end{cases} \quad (5)$$

여기서, 경로의 길이가 2이고(n = 2), 변경된 요소가 두 번째 타입상속 계층이면(i = 2), FTC = 0 이고, BTC = 0 이다. 그리고 확률 pt는 다음과 같이 주어진다.

$$pt = \begin{cases} 1 & \text{if}(i = n) \\ 1 - \left(\frac{(\Pi_{j=i+1}^{n} S_j) - 1}{C_i - 1} \right) & \text{if}(i < n) \end{cases} \quad (6)$$

여기서, C_i는 타입 계층 T_i에서 요소 E_i가 가지는 유일 값의 개수이다. 색인구조의 유지비용은 요소 E_i를 직접 또는 간접으로 참조하는 타겟 타입상속 계층의 요소에 대한 색인 엔트리를 변경전 키 리스트에 해당하는 색인 레코드에서 제거하고, 새로운 키 리스트에 해당하는 색인 레코드에 삽입하는 것이다. 따라서, 이를 IC라 하면 IMC의 평균값은 식 (7)과 같다.

$$IMC = IC \times (1 + pl) \quad (7)$$

pl은 색인 엔트리를 제거할 색인 레코드와 삽입할 색인 레코드가 서로 다른 색인 페이지에 있을 확률이며, IC는 다음 식 (8)과 같이 계산된다.

$$IC = \begin{cases} h + 2 & \text{if } NS \leq P \\ h + 2 + \left(\frac{np - 1}{np} \right) & otherwise \end{cases} \quad (8)$$

여기서, np는 하나의 색인 레코드를 저장하기 위하여 필요한 색인 페이지의 개수이다. pl은 색인 엔트리를 제거할 색인 레코드와 삽입할 색인 레코드가 서로 다른 색인 페이지에 있을 확률이고, pl은 식 (9)와 같다.

$$pl = \begin{cases} 1 & \text{if } NS > P \\ 1 - \left(\frac{NR - 1}{C_n - 1} \right) & otherwise \end{cases} \quad (9)$$

여기서, NR은 색인 페이지당 색인 레코드의 개수이다.

4.1.2 다차원 경로 색인구조의 유지비용

제 3.3절에서 언급한 것처럼 데이터베이스의 변경에 따라 다차원 경로 색인구조를 갱신하기 위해서는 두 번의 순방향 운행과 한번의 색인구조 자체의 갱신이 필요하다. 다차원 중첩요소 색인구조에서와는 달리, 요소 E_{i+1}과 E'_{i+1}에 대한 검색키 값이 서로 동일하여도 색인 레코드에 경로 인스턴스들이 저장되어 있어서 경로 인스턴스의 변경을 반영하기 위해서 색인구조의 변경은 필요하다. 또한 다차원 중첩요소 색인구조에서는 역방향 운행이 필요했으나 다차원 경로 색인구조에서는 역방향 운행이 필요 없다. 따라서 다차원 경로 색인구조의 유지비용 MC는 식 (10)과 같다.

$$MC = 2 \times FTC + IMC \quad (10)$$

여기서, 순방향 운행에 따른 비용인 FTC는 다차원 중첩요소 색인구조에서와 마찬가지이다. 하지만 색인구조의 갱신 비용인 IMC는 다르게 된다. 그 이유는 다차원 경로 색인구조에서는 요소 E_{i+1}과 E'_{i+1}에 대한 검색키 값이 서로 다르고 또한 색인 레코드가 서로 다른 색인 페이지에 있을 경우에만 두 개의 색인 페이지가 갱신되기 때문이다. 따라서 색인구조의 갱신 비용인 IMC는 식 (11)과 같다.

$$IMC = IC \times (1 + pt \times pl) \quad (11)$$

여기서, IC는 식 (8)에서 NS을 PS로 대치한 것과 같다.

4.2 비교 평가

본 절에서는 색인구조를 구축할 경로의 길이에 따른 각 색인구조의 운영에 필요한 유지비용을 비교 평가한다. 각 색인구조의 유지비용은 요소 참조공유도 S_i의 변화에 따라 크게 영향을 받는다. 여기서 S_i는 타입 계층 T_i에서 중첩요소의 값으로 동일한 값을 가지는 요소의 평균 개수이다. 두 색인구조의 유지비용을 비교 평가함에 있어서, 요소 식별자 Eid의 크기는 8바이트, 색인 페이지의 크기 P는 1K바이트로 한다.

4.2.1 경로의 길이가 2인 경우의 유지비용

먼저 색인구조를 구축할 경로의 길이가 2인 경우에 데이터베이스의 변경에 따른 각 색인구조의 갱신을 위한 유지비용에 대한 분석이다. 첫째로, 데이터베이스의 변경이 첫 번째 타겟 타입상속 계층인 T_1에서 발생하는 경우이다. 그림 9는 첫 번째 타겟 타입상속 계층에서 데이터베이스의 변경이 발생한 경우로 첫 번째 타입 계층의 요소 참조공유도 S_1의 값이 10이라 하고, 두 번째 타입 계층의 요소 참조공유도 S_2의 값의 변화에 따른 각 색인구조의 갱신을 위한 유지비용의 변화를 페이지 액세스 수로 나타낸 것이다.

데이터베이스의 변경이 타겟 타입상속 계층 T_1에서 발생하면, 다차원 중첩요소 색인구조에서도 갱신을 위한 역방향 운행에 대한 오버헤드가 필요 없다. 따라서, 다차원 중첩요소 색인구조와 다차원 경로 색인구조의 두 색인구조에서 색인 레코드의 오버플로가 없게 되는 $S_1 \times S_2 < 62$에서는 $MC = (2 \times FTC) + pt \times (BTC + IMC) = 4 + IMC = 4 + IC \times (1 + pl) = 14$로서 일정하다. 그리고 $62 < S_1 \times S_2 < 124$에서는 다차원 경로 색인구조의 색인 레코드에서 오버플로가 발생하므로 유지비용이 1만큼 증가한다. 그러나 $124 < S_1 \times S_2 < 186$에서는 색인 레코드의 개수가 적어짐에 따라 색인구조의 디렉토리의 높이 h가 감소하기 때문에 두 색인구조의 유지비용은 감소한다. $S_1 \times S_2 = 186$인 경우에는 하나의 색인 레코드를 저장하기 위하여 다차원 중첩요소 색인구조에서는 두 개의 색인 페이지가 필요하며, 다차원 경로 색인구조에서는 세 개의 색인 페이지가 필요하다. 따라서 추가적인 페이지 액세스의 확률은 다차원 중첩요소 색인구조에서는 0.5가 되며, 다차원 경로 색인구조에서는 0.7이 된다(참고: IC를 위한 식(8)). 따라서, 다차원 중첩요소 색인구조에서는 IC = 4.5가 되므로 MC = 13이고, 다차원 경로 색인구조에서는 IC= 4.7이 되므로 MC = 13.4이다.

그러나 색인 레코드의 크기가 세 개의 페이지보다 크게 되면, 추가적인 페이지 액세스의 확률이 증가하므로, 디렉토리 높이의 감소에 따른 디렉토리 페이지 액세스의 감소에 대한 이득을 상쇄하게 된다. 따라서 $186 < S_1 \times S_2$ 인 경우에는 두 색인구조에서 모두 유지 비용이 14에 가깝게 된다.

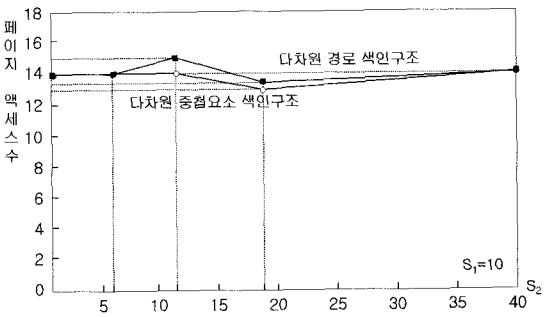


그림 9. 경로의 길이가 2인 경로상의 첫 번째 타입상속 계층에서 데이터베이스의 변경이 발생한 경우 두 색인구조의 갱신을 위한 유지비용의 변화

둘째로, 데이터베이스의 변경이 두 번째 타입 계층인 T_2에서 발생하는 경우이다. 이 경우에는 역방향 운행과 순방향 운행이 모두 필요 없기 때문에 요소 참조공유도 S_i가 매우 낮으면(S_1, S_2 모두 1에 근접할 때) 두 색인구조의 유지비용이 같아진다. 요소 참조공유도 S_i가 증가하면, 다차원 중첩요소 색인구조와 다차원 경로 색인구조의 유지비용은 데이터베이스의 변경이 타겟 타입상속 계층 T_1에서 발생할 경우와 같은 경향을 보인다.

따라서 위의 결과를 분석하면, 색인구조를 구축할 경로의 길이가 2인 경우에는 다차원 중첩요소 색인구조가 적합함을 알 수가 있다. 이것은 두 색인구조 모두 역방향 운행이 필요 없기 때문에 유지비용은 비슷하지만 다차원 경로 색인구조는 색인 레코드의 크기에 의한 저장 공간의 오버헤드로 인하여 검색성능이 떨어지지 때문이다.

4.2.2 경로의 길이가 3인 경우의 유지비용

다음으로, 색인구조를 구축할 경로의 길이가 3인 경우에 데이터베이스의 변경에 따른 각 색인구조의 갱신을 위한 유지비용에 대한 분석이다. 경로 길이가 2보다 크면, 다차원 중첩요소 색인구조에서는 제 3.4절에서 제시한 바와 같이 한 번의 역방향 운행이 필요하게 되므로 다차원 경로 색인구조에 비해 유지비용이 증가한다.

첫째로, 데이터베이스의 변경이 세 번째 타입 계층 T_3에서 발생하는 경우이다. 그림 10은 요소 참조공유도 S_2, S_3의 값을 5라 하고, S_1의 변화에 따른 각 색인구조의 갱신을 위한 유지비용의 변화를 나타낸 것이다. 다차원 중첩요소 색인구조의 유지비

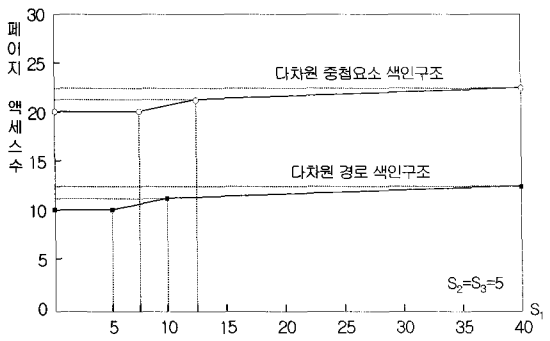


그림 10. 경로의 길이가 3인 경로상의 세 번째 타입상속 계층에서 데이터베이스의 변경이 발생한 경우 두 색인구조의 갱신을 위한 유지비용의 변화

용을 지배하는 것은 역방향 운행에 의한 것으로, 역방향 운행에 필요한 요소의 액세스 개수는 S_2개이다(참고: BTC를 위한 식(5)). 예를 들어, $S_2 = S_3 = 5$인 경우 다섯 개의 요소가 반드시 액세스되어야 하며, 각 요소에 대해 두 번의 페이지 액세스가 필요하므로 역방향 운행에 필요한 비용은 10번의 페이지 액세스가 된다. 그리고 두 색인구조 모두 유지비용이 S_1의 증가에 따라 증가함을 보인다. 이것은 색인 레코드의 크기가 페이지의 크기보다 커지기 때문에 추가적인 페이지 액세스의 가능성이 높아지기 때문이다.

둘째로, 데이터베이스의 변경이 첫 번째 타겟 타입상속 계층 T_1과 두 번째 타입 계층 T_2에서 발생하는 경우이다. 이 경우는 다차원 중첩요소 색인구조에서도 역방향 운행이 필요 없으므로 색인구조의 유지비용이 다차원 경로 색인구조에서와 같다.

따라서 위의 결과를 분석하면, 색인구조를 구축할 경로의 길이가 3인 경우에는 다차원 경로 색인구조가 적합함을 알 수가 있다. 이것은 다차원 경로 색인구조의 검색 성능은 다차원 중첩요소 색인구조와 유사한 반면에 데이터베이스의 변경에 의한 유지비용이 적게 들기 때문이다. 그러나 데이터베이스의 변경이 첫 번째와 두 번째 타입 계층에서 주로 발생하고, 각 타입 계층의 요소 참조공유도의 곱이 크게 되면 다차원 중첩요소 색인구조가 적합한 색인구조이다. 이러한 경우에는 다차원 중첩요소 색인구조에서도 갱신을 위한 역방향 운행이 필요 없게 되므로 색인구조의 유지비용이 다차원 경로 색인구조에서와 같게 되기 때문이다.

4.2.3 경로의 길이가 4이상인 경우의 유지비용

경로의 길이가 n인 일반적인 경우, 다차원 중첩요소 색인구조와 다차원 경로 색인구조의 유지비용에 지배적인 영향을 미치는 것은 데이터베이스의 경로를 따른 순방향 운행과 역방향 운행이다. 즉, 다차원 경로 색인구조의 유지비용은 경로의 길이에 비례하게 되며, 다차원 중첩요소 색인구조의 유지비용은 경로 길이와 참조 공유도 S_i들의 곱에 비례하게 된다. 이것은 데이터베이스의 변경이 경로를 따라 모든 클래스 계층에서 동등하게 발생한다고 가정하고 구하는 평균 유지비용의 계산식으로 알 수 있다.

즉, 다차원 경로 색인구조에서는 두 번의 순방향 운행이 필요하므로, *FTC*에 관한 식(3)으로부터 평균 유지비용 A는

$$A = 2 \times [2 \times \sum_{i=1}^{n}(n-i)]/n = 2 \times (n-1)$$

이다. 따라서 유지비용은 경로의 길이 n에 비례함을 알 수 있다. 그리고 다차원 중첩요소 색인구조에서는 순방향 운행과 더불어 역방향 운행이 필요하므로, *BTC*에 관한 식(5)로부터 역방향 운행의 평균 비용 R은

$$\begin{aligned} R = [2 \times (&S_{n-1} \times S_{n-2} \times \cdots \times S_2 \\ &+ S_{n-2} \times S_{n-3} \times \cdots \times S_2 + \cdots \\ &+ S_3 \times S_2 + S_2)] \,/n \end{aligned}$$

이다. 따라서 다차원 중첩요소 색인구조의 평균 유지비용 B은 이 비용에 다차원 경로 색인구조에서와 같은 두 번의 순방향 운행에 필요한 비용을 더하여 B = A + R이 된다.

이와 같은 유지비용에 관한 식으로부터 경로의 길이가 4이상인 경우에는 두 색인구조 모두 사용하기가 어렵다. 즉, 다차원 중첩요소 색인구조에서는 요소 참조공유도 S_i에 비례해서 증가하는 역방향 운행의 유지비용이 매우 높게 되기 때문에 데이터베이스의 변경이 거의 발생하지 않거나, 요소 참조공유도가 매우 낮은 경우가 아니면 사용이 불가능하게 된다. 그리고 다차원 경로 색인구조에서는 색인구조에서 경로 인스턴스(Eid들의 리스트)들을 유지하기 때문에 요소 참조공유도 S_i가 증가함에 따라 색인 레코드가 매우 커져서 여러 개의 페이지를 점유하게 되어 검색비용이 증가하여, 요소 참조공유도가 매우 낮은 경우가 아니면 사용이 불가능하게 된다.

따라서, 경로의 길이가 4이상인 경우에는 경로를 길이가 1, 2 또는 3이 되는 서브경로들로 분할한 다음, 각 서브경로에 따라 다차원 중첩요소 색인구조 또는 다차원 경로 색인구조로 선택적으로 할당하여 사용하여야 한다.

5. 결 론

본 논문에서는 XML 데이터베이스의 중첩요소에 대한 색인기법으로, 기존의 색인기법으로는 지원할 수 없는 타입상속 계층과 중첩요소가 포함된 복합형태의 질의처리를 지원하기 위하여 다차원 색인구조를 이용하는 다차원 타입상속 색인기법인 MD-TIX를 제안하였다. MD-TIX에서는 색인된 중첩요소의 킷값 도메인과 함께 중첩요소를 표현하는 경로상의 각 타입상속 계층마다 한 축의 타입식별자 도메인을 할당하여 구성된 다차원 도메인 공간상에 주어진 색인 엔트리들의 클러스터링을 다루는 색인기법이다. 제안된 색인기법은 질의의 대상 범위가 타입상속 계층상의 임의의 타입들로 제한되거나, 질의에 포함된 복합요소들의 도메인이 타입상속 계층상의 임의의 타입들로 제한되는 경우에도 효율적으로 지원할 수 있다.

또한, 본 논문에서는 중첩요소에 대한 색인구조로서 다차원 파일구조의 하나인 계층 그리드 파일을 이용하여, 색인 엔트리를 타겟 타입상속 계층의 요소식별자(즉, Eid)들로 구성하는 다차원 중첩요소 색인구조와 색인 엔트리를 색인된 중첩요소에 대한 경로 인스턴스(즉, Eid 리스트)들로 구성하는 다차원 경로 색인구조의 두 가지 형태의 색인구조를 제안하였다. 그리고 이들의 성능평가를 위하여 데이터베이스의 변경에 따른 각 색인구조의 갱신을 위한 운영 알고리즘과 각 색인구조의 운영에 따른 유지비용을 비교 평가하였다. 평가 결과로서 경로의 길이가 2인 경우에는 데이터베이스의 변경에 따른 색인구조의 갱신 비용은 두 색인구조 모두 14 페이지 액세스 정도로 비슷한 반면에 저장 공간의 오버헤드가 적은 다차원 중첩요소 색인구조를 구축하는 것이 합당하고, 경로의 길이가 3인 경우에는 색인구조의 갱신을 위해서 색인된 경로에 대한 역방향 운행에 필요한 10 페이지 액세스 정도의 비용이 더 필요한 다차원 중첩요소 색인구조에 비해 역방향 운행이 필요 없는 다차원 경로 색인구조를 구축하는 것이 합당한 것으로 나타났다. 그리고 경로의 길이가 4이상일 경우에는 경로의 길이에 따라 증가하는 색인구조의 유지비용으로 인하여, 경로의 길이가 1, 2, 또는 3이 되는 서브경로로 나누어서 각 서브경로별로 적합한 색인구조를 할당하여야 함을 알 수 있었다.

참 고 문 헌

[1] T. Bray et al., *Extensible Markup Language, (XML) 1.0. W3C Recommendation*, http://www.w3.org/TR/REC-xml-19980210, Feb. 2004.

[2] W. Meier, "eXist: An Open Source native XML Database," *Web, Web-Services, and Database Systems, NODe 2002 Web- and Database-Related Workshops*, Revised Papers (Lecture Notes in Computer Science Vol. 2593), pp. 169-183, 2003.

[3] C. D. Fallside and P. Walmsley, *XML Schema Part 0. W3C Recommendation*, http: //www.w3.org /TR /xmlschema-0, Oct. 2004.

[4] S. Finkelstein et al., "Physical Database Design for Relational Databases," *ACM Trans. on Database Systems*, Vol.13, No.1, pp. 91-128, Mar. 1988.

[5] A. Berglund et al., "XML Path Language (XPath) 2.0. W3C Working Draft 30 Apr. 2002," *http://www.w3.org /TR /xpath20,* Working Draft, 2002.

[6] R. Goldman and J. Widom, "DataGuides: Enable Query Formulation and Optimization in Semistructured DataBases," In *Proc. Int'l Conf. on Very Large Data Bases*, Athens, Greece, pp. 436-445, Aug. 1997.

[7] T. Milo and D. Suciu, "Index Structures for Path Expression," In *Proc. Int'l Conf. on Database Theory*, Jerusalem, Israel, pp. 277-295, Jan. 1999.

[8] B. F. Cooper et al., "A Fast Index for Semistructured Data," In *Proc. Intl. Conf. on Very Large Data Bases*, Rome, Italy, pp.

341-350, Sept. 2001.

[9] C. W. Chung, J. K. Min, and K. Shim. "APEX: An Adaptive Path Index for XML Data," In *Proc. Intl. Conf. on Management of Data, ACM SIGMOD*, Madison, Wisconsin, pp. 121-132, June, 2002.

[10] S. Nestorov et al., "Representative Objects: Concise Prepresentation of Semistructured, Hierarchical Data," In *Proc. IEEE Int'l Conf. on Data Engineering*, Birmingham, U.K., pp. 79-90, Feb. 1997.

[11] D. Comer, "The Ubiquitous B-tree," *ACM Computing Surveys*, New York, USA, Vol.11, No.2, pp. 121-137, June 1979.

[12] J. H. Lee et al., "A Region Splitting Strategy for Physical Database Design of Multidimensional File Organizations," In *Proc. Int'l Conf. on Very Large Data Bases*, Athens, Greece, pp. 416-425, Aug. 1997.

[13] J. L. Bentley, "Multidimensional Binary Search Trees in Database Applications," *IEEE Trans. on Software Eng.*, Vol.5, No.4, pp. 333-340, July 1979.

[14] J. T. Robinson, "The K-D-B-Tree: A Search Structure for Large Multidimensional Dynamic Indexes," In *Proc. Int'l Conf. on Management of Data, ACM SIGMOD*, Ann Arbor, Michigan, pp. 10-18, Apr. 1981.

[15] D. Lomet and B. Salzberg, "The hB-tree: A Multiattribute Indexing Method with Good Guaranteed Performance," *ACM Trans. on Database Systems*, Vol.15, No.4, pp. 625-658, Dec. 1990.

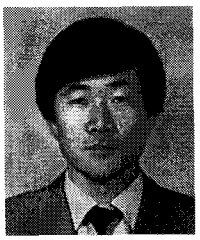
[16] J. Nievergelt et al., "The Grid File: An Adaptable, Symmetric Multikey File Structure," *ACM Trans. on Database Systems*, Vol.9, No.1, pp. 38-71, Mar. 1984.

[17] K. Y. Whang and R. Krishnamurthy, "The Multilevel Grid File - A Dynamic Hierarchical Multidimensional File Structure," In *Proc. Intl. Conf. on Database Systems for Advanced Applications(DASFAA)*, Tokyo, pp. 449-459, Apr. 1991.

[18] J. H. Lee, "2D-THI: Two-Dimensional Type Hierarchy Index for XML Databases," *Journal of Korea Multimedia Society*, Vol.9, No.3, pp. 265-278, Mar. 2006.

[19] S. Boag et al., *XQuery 1.0: An XML Query Language*, http://www.w3.org/TR/xquery, Nov. 2005.

[20] E. Horowitz and S. Sahni, *Fundamentals of Computer Algorithms*, Computer Science Press, Potomac, Maryland, 1978.



이 종 학

1982년 경북대학교 전자공학과(전자계산 전공) 졸업(학사)
1984년 한국과학기술원 전산학과 졸업(공학석사)
1997년 한국과학기술원 전산학과 졸업(공학박사)
1991년 정보처리기술사
1984년~1987년 금성통신(주) 부설연구소 주임연구원
1987년~1998년 한국통신 연구개발본부 선임연구원
1998년~현재 대구가톨릭대학교 컴퓨터정보통신공학부 교수
관심분야 : 객체관계형 데이터베이스, 다차원 파일구조, 물리적 데이터베이스 설계, XML 데이터베이스, 데이터 웨어하우스, 생물정보학 등