Journal of Korea Multimedia Society Vol. 15, No. 9, September 2012(pp. 1133-1148)

조상-자손 관계 탐색을 지원하기 위한 XML 타입상속 색인구조의 계층적 구성기법

이 종 학†

요 약

본 논문에서는 XML 데이터베이스에서 XML 질의처리를 효율적으로 지원하기 위한 다차원 타입상속 색인구조(MD-TIX)들의 계층적 구성기법을 제시한다. MD-TIX는 중첩요소와 여러 타입상속 계층으로 이루어진 중첩술어의 조상-자손 관계 탐색을 효율적으로 지원하기 위하여 다차원 색인구조를 이용하는 색인기법이다. 그러나 이러한 MD-TIX는 질의에 주어진 Xpath의 길이가 긴 경우에 색인 엔트리의 구성문제 때문에 색인구조의 구축과 유지관리에 어려움이 있다. 이를 극복하기 위해서, 본 논문에서는 먼저 주어진 Xpath에서 인접한 두 타입 사이의 직접 관계 탐색을 지원하는 기본 색인구조들을 구축하고, 이들을 바탕으로 Xpath상의 임의의 두 타입 사이의 간접 관계 탐색을 지원하는 유도 색인구조들을 구축한다. 이러한 과정을 전체 길이의 Xpath를 지원하는 하나의 목표 색인구조를 구축하기까지 계층적으로 구성하는 방법을 제시한다. 또한 Xpath 상에 주어진 몇 개의 부경로 탐색만을 지원하기 위한 부분적인 색인계층을 효율적으로 구축하기 위한 알고리즘을 제안한다.

XML Type Inheritance Index Hierarchies for Supporting Ancestor-descendant Exploration

Jong-hak Lee†

ABSTRACT

This paper presents a hierarchical structuring method for the multidimensional type inheritance indices (MD-TIXs) that support the processing of XML queries in XML databases. MD-TIX uses a multidimensional index structure for efficiently supporting ancestor-descendant explorations that involve both nested element and type inheritance hierarchies. However, In the case of a long Xpath, the building and maintenance of MD-TIX are very difficult because of index entry construction problem. So, we propose a type inheritance index hierarchy method for solving this difficulty. We first construct base indices that support direct relationship explorations between adjacent two types on a given Xpath, and then, based on these base indices, we construct hierarchically the derived indices that support indirect relationship explorations between any two types of Xpath until constructing one target index for supporting the full Xpath. And we also present an algorithm that efficiently constructs a partial index hierarchy for supporting given a set of sub-pathes explorations.

Key words: XML Databases(XML 데이터베이스), XML Schema(XML 스키마), XML Query Processing (XML 질의처리), XML Index(XML 색인구조)

※ 교신저자(Corresponding Author) : 이 종학, 주소 : 경북 경산시 하양읍 금락1리 330(712-702), 전화 : 053)850-2746, FAX : 053)850-2750, E-mail : jhlee11@cu.ac.kr
접수일 : 2012년 4월 30일, 수정일 : 2012년 7월 10일
완료일 : 2012년 8월 6일
† 정회원, 대구가톨릭대학교 IT공학부 교수
※ 이 논문은 2011년도 대구가톨릭대학교 교내연구비 지원에 의한 것임.

1. 서 론

XML[1] 데이터베이스 관리 시스템의 질의처리 성능 최적화는 중요한 연구과제이다. 최근에 제안된 XML 데이터베이스 색인기법들은 XML 질의처리의 성능 향상에 크게 기여하고 있다[2,3]. 그러나 이들 색인구조들은 기존의 관계형 데이터베이스의 단순 속성에 대한 색인구조에 비해 저장 공간 및 갱신유지 비용에 큰 부담이 있다. 또한, 색인구조의 종류에 따라 검색 성능이 다른 특성도 있다[2,4]. 그러므로 XML 데이터베이스 색인기법을 통한 질의처리의 이점이 색인구조의 저장 공간 및 갱신유지를 위한 부담으로 인해 상쇄되지 않기 위해서는 색인구조들을 매우 신중히 할당하고 구축하여야 한다[4].

XML 데이터베이스[5]에서 XML 스키마는 타입 집단화(type aggregation) 개념에 의하여 한 타입이 가지는 요소(element)의 도메인이 또 다른 타입이 될 수 있도록 함으로써(이러한 요소를 *복합요소(complex element)*라 함) 타입들 사이에 타입 집단화 계층을 이룬다. 따라서 타입 집단화 계층을 이루는 타입들에서 정의된 어떠한 요소도 논리적으로는 루트 타입의 요소라고 볼 수 있다. 본 논문에서는 타입 집단화 계층에서 루트 타입이 아닌 타입에서 정의된 요소를 루트 타입의 *중첩요소(nested element)*[6,7]라 한다. 이로 인하여 XML 질의어에서는 중첩요소에 조건이 주어지는 *중첩술어(nested predicate)*[8]를 가진다는 특징이 있다. XML 질의어에서는 중첩요소를 표현하기 위해서 Xpath[6]와 같은 경로 표현식을 사용한다. 경로 표현식은 루트 타입으로부터 타입 집단화 계층을 따라 나타나는 요소들의 나열로 표현한다.

XML 스키마는 또 하나의 중요한 개념인 타입 상속(type inheritance) 개념에 의하여 타입들 사이에 타입상속 계층이라는 또 다른 계층구조를 이룬다. 즉, 하나의 타입은 여러 개의 서브타입들을 가지며, 각 서브타입은 또 다른 여러 서브타입들을 가진다[9]. 이로 인하여 XML 데이터베이스에서는 하나의 질의에 대한 대상 범위를 두 가지 경우로 해석할 수 있다. 한 경우는 질의의 대상 범위를 질의에 나타나는 타입만으로 한정하는 것이고, 또 다른 경우는 질의의 대상 범위를 질의에 나타나는 타입과 그의 모든 서브타입들을 포함하는 것이다. 이러한 타입상속 계층에 대한 질의를 효율적으로 처리할 수 있는 색인구조는 특정 타입에 속하는 요소(element)들의 탐색뿐만 아니라 특정 타입을 루트로 하는 타입상속 계층의 모든 타입들에 속하는 요소들의 탐색도 효율적으로 처리할 수 있어야 한다.

타입 집단화 계층에서 정의된 중첩요소를 표현하는 Xpath에는 요소들 사이의 중첩관계에 의한 요소와 요소 사이에 *암시적 조인(implicit join)*[10]의 의미를 가지고 있다. 이러한 암시적 조인은 XML 스키마에 의해 미리 예상이 가능하다. 따라서 질의에 자주 나타나는 중첩요소에 대한 암시적 조인을 미리 계산하여 그 결과를 색인으로 구축하여 놓음으로써, 질의처리시 이를 이용하여 성능 향상을 꾀할 수 있으며 이를 중첩요소에 대한 색인기법[2,7,11-14]이라 한다. 그러나 이러한 중첩요소에 대한 색인기법들은 대부분 일차원 색인구조인 B-tree를 이용함으로써, 타입 상속의 특징으로 인한 질의의 대상 범위가 타입상속 계층상의 임의의 타입들로 제한되거나, Xpath에 나타나는 요소의 도메인이 타입상속 계층상의 임의의 타입들로 제한이 되는 질의들을 지원하기 어려운 문제점을 가지고 있다.

이와 같은 일차원 색인구조를 이용하는 중첩요소에 대한 색인기법들이 가지는 타입상속 계층의 지원 문제를 해결하기 위하여, 다차원 색인구조[15,16]를 중첩요소에 대한 색인구조로 이용할 수 있으며 이를 *다차원 타입상속 색인구조(MD-TIX: Multidimensional Type Inheritance indeX)*[17]라 한다. MD-TIX에서는 중첩요소의 키 값 도메인과 함께, Xpath에 나타나는 루트 타입의 타입상속 계층과 각 복합요소의 도메인 타입상속 계층마다 한 축의 타입 식별자 도메인을 할당하여 다차원 색인구조를 구성한다.

하지만 이러한 MD-TIX는 질의에 주어진 Xpath의 길이가 긴 경우에 색인구조의 구축과 데이터베이스의 갱신에 따른 색인구조의 유지관리에 많은 어려움이 있다[17]. 이는 색인키와 목표요소 사이의 관계를 나타내는 색인 엔트리를 구성하기 위하여 많은 데이터베이스의 탐색이 필요하기 때문이다. 이러한 어려움 때문에 참고문헌[17]에서는 색인 엔트리를 경로 길이만큼의 요소 식별자들의 리스트로 구성하는 다차원 경로 색인구조를 제안하였다. 이는 또한 경로의 길이에 따라 증가하는 색인구조의 막대한 저장비용으로 인하여, 길이가 4이상인 경로에서는 색

인구조의 구축이 어렵다[17].

따라서 본 논문에서는 이러한 문제를 해결하기 위하여 데이터베이스의 탐색 없이도 색인엔트리를 색인키와 목표요소만으로 구성할 수 있도록 색인구조들을 계층적으로 구성하는 방안을 제안한다. 즉, 먼저 주어진 길이가 n인 Xpath에서 인접한 두 타입 사이의 직접 관계 탐색을 지원하는 n-1개의 기본 색인구조들을 구축하고, 이들을 바탕으로 Xpath상의 임의의 두 타입 사이의 간접 관계 탐색을 지원하는 유도 색인구조들을 구축한다. 이러한 과정을 전체 길이의 Xpath를 지원하는 하나의 목표 색인구조를 구축하기까지 계층적으로 구성한다. 또한 Xpath상에서 탐색의 목표로 주어진 몇 개의 부경로 탐색만을 지원하기 위한 부분 색인계층을 효율적으로 구축하고 운영하기 위한 보조 색인구조들을 선택하기 위한 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 먼저, 제 2절에서는 관련 연구로서 XML 데이터베이스의 색인 구축에 필요한 기본 개념으로 XML 질의어와 질의처리 전략 및 기존의 색인 기법들을 살펴본다. 제 3절에서는 Xpath 상에서 인접한 두 개의 타입들 사이에 구축하는 기본 색인구조들을 소개하고, 이들로부터 전체 길이의 Xpath를 지원하는 하나의 목표 색인구조를 계층적으로 구축하는 방법을 제시한다. 제 4절에서는 부분 색인계층의 구축을 위한 보조 색인구조들을 선택하기 위한 알고리즘을 제안한다. 그리고 제 5절에서는 기존의 다차원 경로 색인구조와 함께 본 논문에서 제안한 계층적 색인구조들 사이의 성능평가 결과를 제시한다. 마지막으로 제 6절에서는 결론을 내린다.

2. 관련 연구

본 절에서는 XML 데이터베이스의 타입상속 색인구조의 계층적 구성기법에 필요한 기본 개념들을 기술한다. XML 데이터베이스는 XML 문서를 효율적으로 저장하고 검색하기 위하여 폭넓은 연구가 진행되고 있으며, XML 스키마(Schema)[9], XQuery[8], XPath[6] 등과 같은 XML 관련기술에 대한 표준이 제안되어 있다. 본 절에서는 먼저 XML 스키마에 대하여 소개한 다음, XPath와 함께 XML 질의어의 특징, 그리고 XML 질의처리 전략 및 기존의 XML 색인기법들에 관하여 기술한다.

2.1 XML 스키마

XML 스키마는 XML 문서의 구조를 정의하기 위하여 제안된 XML 문서 정의어이다[9]. 그림 1은 Persons 타입의 문서를 정의하는 XML 스키마 그래

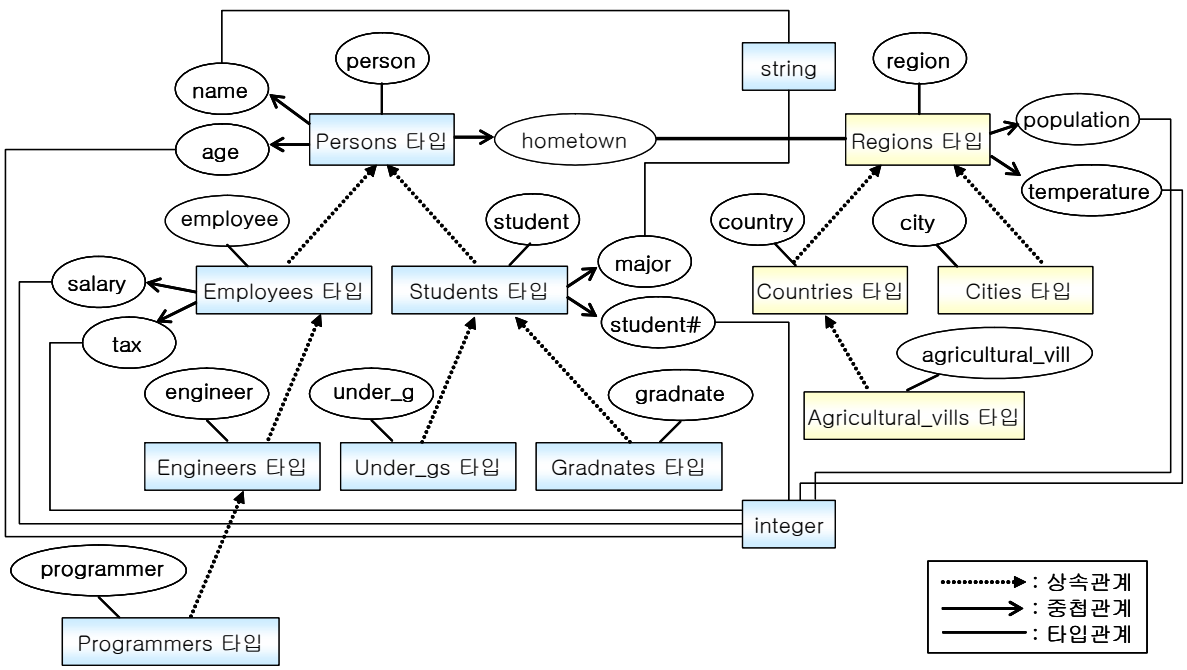


그림 1. *Persons* 타입에 대한 XML 스키마 그래프

프의 예를 나타낸다. 그림에서 타입은 네모로 나타내고, 타입을 구성하는 요소는 동그라미로 나타내며, 타입 간의 상속관계는 점선 화살표로 나타낸다. 그리고 요소와 타입 간의 중첩관계를 실선 화살표로 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 타입 상속 계층구조와 복합요소인 hometown의 도메인 타입인 Regions 타입을 포함하는 XML 타입 집단화 계층구조의 루트이다.

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, Under_gs 타입, Graduates 타입, Programmers 타입}이고, $Students^*$ 타입은 {Students 타입, Under_gs 타입, Graduates 타입}이다.

2.2 XML 질의어

XML 질의어에서는 타입 집단화 계층구조에서 중첩요소의 경로를 표현하기 위하여 Xpath라는 하나의 경로식(path expression)을 사용한다[1]. 본 논문에서는 경로식에서 경로를 이루는 요소들의 타입을 타입상속 계층상의 일부 타입들로 한정(limit)하여 표현할 수 있도록 XPath를 확장하여 이를 *확장된* XPath라 한다. 확장된 XPath는 각 요소 다음에 한정된 타입 이름들이 ( )속에 올 수 있도록 확장한 것으로 다음과 같은 형태를 가진다. 단, E_i 뒤의 중괄호 { }는 선택적임을 나타내는 표시이다.

$$XP = T_0/E_1\{(T_1)\}/E_2\{(T_2)\}/\ldots/E_n\{(T_n)\} \quad (1)$$

경로 *XP*에서 타입 T_0을 *타겟타입*, T_i을 요소 E_i의 *도메인타입*이라 한다. 타겟타입과 도메인타입은 경로에서 타입상속 계층구조에 속하는 특정 타입으로 한정될 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.

확장된 XPath식 *XP*에서 *경로 인스턴스(path instance)*는 다음 조건을 만족하는 요소들의 리스트 (E_0, E_1, ..., E_n)로 정의한다. (1) 요소 E_0은 타입 T_0의 요소이다. (2) 요소 E_i ($0 < i <= n$)는 타입 T_i의 요소로서 요소 E_{i-1}의 구성 요소이다. 그리고 확장된 XPath식 *XP*에서 E_i와 E_j ($j >= i+1$)는 조상-후손(ancestor-descendant) 관계이고, 이들 중 $j = i+1$이면 이들은 부모-자식(parent-child) 관계이다.

2.3 질의처리 전략 및 기존 색인기법

XML 데이터베이스의 중첩술어를 가지는 질의를 처리하기 위한 스키마 그래프의 운행 전략으로 순방향 운행(FT: Forward Traversal) 전략과 역방향 운행(BT: Backward Traversal) 전략이 있다[18]. FT 전략은 스키마 그래프의 집단화 계층구조에서 상위 타입을 먼저 탐색하고 하위 타입을 나중에 탐색하는 방법으로 깊이-우선 순서로 집단화 계층구조를 운행한다. 그리고 BT 전략은 하위 타입을 먼저 탐색하고 상위 타입을 나중에 탐색하는 방법으로 집단화 계층구조를 아래에서 위로 운행한다. 역방향 운행시에는 하위 타입의 요소를 참조하는 상위 타입의 요소를 탐색하기 위해서는 많은 비용을 필요로 하기 때문에 색인구조를 구축하여 이를 이용하게 된다.

XML 데이터베이스에서 색인을 유지하는 요소가 단순요소이면 관계형 데이터베이스 시스템에서와 같이 색인된 요소에 대한 제한 술어를 만족하는 요소의 신속한 탐색을 위해서 사용될 수 있고, 색인을 유지하는 요소가 복합요소이면 역방향 운행시 상위 타입의 요소를 탐색하기 위한 비용을 줄일 수 있다. 따라서 XML 데이터베이스에서 색인의 키 값은 단순요소의 도메인이 되는 기본 타입의 값들뿐만 아니라 복합요소의 도메인이 되는 사용자 정의 타입의 요소 식별자인 Eid들도 될 수 있다. 그리고 중첩요소에 대한 색인기법으로 중첩요소를 표현하는 Xpath에 의한 암시적 조인을 미리 계산하여 색인구조로 유지함으로써, 질의처리시 타입 집단화 계층에 대한 운행을 생략할 수도 있다[11,13].

지금까지 제안된 XML 데이터베이스의 중첩요소에 대한 색인기법으로는 Index Fabric[7], APEX[2], DataGuide[11], 1-Index[13], SphinX[12], Twig Pattern Matching[14]등이 있다. DataGuide는 비결정적 오토마타를 결정적 오토마타로 변환하는 과정과 동일한 과정으로 경로를 색인하는 기법이다. 일반적으로 비결정적 오토마타를 결정적 오토마타로 바꿀 경우, 크기가 커지게 되지만, XML 문서 내에 동일

한 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. Twig Pattern Matching은 두 개 이상의 술어로 구성된 가지 패턴의 질의를 지원하기 위한 색인기법이다.

이러한 기존의 색인기법들은 일차원 색인구조인 B-tree를 이용함으로써 XML 데이터 모델의 타입상속의 특징을 반영하지 못하는 것들로서, 타겟타입의 대치 또는 도메인타입의 대치가 있는 질의는 지원하지 못하는 한계점이 있다. 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. 따라서 이러한 중첩 술어의 처리를 지원하기 위한 색인구조로 다차원 색인구조[15,16]를 이용할 수 있다. 즉, 색인할 중첩요소의 킷값 도메인과 함께 중첩요소를 표현하는 경로상의 각 타입상속 계층마다 타입식별자들로 구성된 한 차원씩의 타입식별자 도메인을 할당함으로써, (*경로길이+1*)차원의 도메인 공간을 구성하여 이를 다차원 색인구조에 적용한다.

참고문헌[17]에서는 중첩요소에 대한 다차원 타겟요소 색인구조를 다차원 파일구조의 하나인 계층 그리드 파일(MLGF: MultiLevel Grid File)[16]을 이용하여 구성하고, 이를 *다차원 타입상속 색인구조*(MD-TIX)라 하였다. 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 크게 될 때 오버플로우 페이지를 할당하고 이를 포인트하기 위한 오버플로우 페이지(overflow page) 필드가 있다.

다차원 타입상속 색인구조는 색인 레코드에 있는 색인 엔트리의 구성방법에 따라 *다차원 타겟요소 색인구조(MD-TEI: Multidimensional Taget Element Index)*와 *다차원 경로 색인구조(MD-PI: Multidimensional Path Index)*의 두 가지 색인구조로 분류한다. MD-TEI는 색인 엔트리를 색인된 중첩요소의 타겟 타입상속 계층에 속하는 요소에 대한 요소 식별자(즉, Eid)들로 구성하며, MD-PI는 색인 엔트리를 색인된 중첩요소에 대한 경로 인스턴스(즉, Eid 리스트)들로 구성한다. 그림 2는 이러한 MD-PI의 색인 페이지 구조를 나타낸다.

참고문헌[17]에서는 MD-TEI의 경우 검색 성능은 아주 좋지만, Xpath의 길이가 3이상이 되면 색인구조의 구축이 어렵고 유지비용이 너무 크게 되어 색인구조 운영의 어려움을 밝히고 있다. 이는 데이터베이스의 갱신이 있을 경우 색인 엔트리의 재구성을 위한 경로상의 역방향 운행이 필요하게 되고, 데이터베이스의 각 요소가 역 참조자를 갖지 않을 경우 많은 데이터베이스 탐색비용을 필요로 하기 때문이다. 따라서 본 논문에서는 Xpath의 길이가 3이상이 되는

그림 2. MD-PI의 색인 페이지 구조

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

경우에도 색인 엔트리의 재구성을 위한 경로상의 역방향 운행과 같은 데이터베이스 탐색을 하지 않고, MD-TEI의 구축이 가능할 수 있도록 여러 개의 MD-TEI들을 계층적으로 구성하는 방안을 제안한다.

3. 다차원 타겟요소 색인구조의 계층적 구성

중첩요소에 대한 색인기법은 경로상의 참조관계에 의한 암시적 조인의 연산을 미리 계산하여 두는 기법으로 중첩술어를 만족하는 요소들의 탐색에는 매우 유용하지만, 상대적으로 색인구조의 유지비용을 많이 필요로 한다. 이러한 색인구조의 유지비용은 색인을 유지하는 경로의 길이에 비례하기 때문에, 경로의 길이가 매우 길 때는 경로를 여러 개의 부경로(sub-path)로 나누어서, 부경로별로 여러 개의 색인구조를 유지함으로써 유지비용을 줄일 수 있다. 특히, 길이가 2인 부경로에 대한 색인구조는 이차원 타겟요소 색인구조(2D-TEI: Two-Dimensional Taget element Index)[19]가 된다. 즉, 경로상의 모든 중첩요소들에 대해서 이차원 타겟요소 색인구조를 구축함으로써 중첩술어를 만족하는 타겟 타입상속 계층의 요소들을 여러 번의 단계적인 색인 검색으로 구할 수 있다. 우리는 이와 같이 경로상의 중첩요소마다 구축하는 색인구조를 *기본 타겟요소 색인구조(B-TEI: Base Taget Element Index)*라 한다.

그리고 인접한 두 B-TEI를 병합함으로써 길이가 3인 부경로들에 대한 타겟요소 색인구조를 구성할 수 있고, 이런 인접한 타겟요소 색인구조의 병합을 단계적으로 적용함으로서 최종적으로 길이가 n인 경로에 대한 하나의 타겟요소 색인구조를 구성할 수 있다.

XML 스키마의 한 경로는 노드가 타입상속 계층을 나타내고 에지가 타입들 사이의 관계를 나타내는 방향성 그래프로 표현할 수 있다. E_i는 타입상속 계층 T_i의 요소이고 E_i 값의 범위가 타입상속 계층 T_j라 하면, 스키마 그래프에서는 T_i 에서 T_j 로의 방향성 에지가 있게 되며 E_j 라는 라벨을 갖는다. 더 나아가서 $i = 0, 1, \ldots, n-1$에 대해서, E_i 라벨을 갖는 T_i에서 T_{i+1}로의 방향성 에지가 있으면 $<T_0, E_1, T_1, E_2, \cdots, E_n, T_n>$은 하나의 경로 스키마를 나타낸다. 다음의 그림 3은 길이가 5인 XML 경로 스키마를 나타낸다.

XML 경로 스키마 $<T_0, E_1, T_1, E_2, \cdots, E_n, T_n>$에 대해, 타겟요소 색인구조 TEI$(i, j)$ $(1 \le i < j \le n)$는 $(Eid(e_i), Tid(e_i), Eid(e_j), m)$로 구성된 색인 레코드들의 집합으로 구성된다. 여기서 e_i와 e_j는 각각 타입계층 T_i와 T_j의 요소이며, $k = 0, 1, \ldots, j-i-1$에 대해서, e_{i+k+1}는 e_{i+k}에 의해서 참조되는 요소 경로 $<e_i, e_{i+1}, \ldots, e_{j-1}, e_j>$가 존재하고 m은 e_i와 e_j에 연결된 서로 다른 요소 경로의 개수를 나타낸다. 즉 $Eid(e_i)$는 타겟요소의 식별자이고, $Tid(e_i)$는 타겟요소가 속하는 타입의 타입 식별자로서 색인구조를 구성하는 한 차원의 도메인을 구성하고, $Eid(e_j)$는 색인 킷값으로 색인구조의 또 다른 한 차원의 도메인을 구성한다.

경로 스키마를 따라서 서로 다른 두 타입의 요소들 사이의 직접 또는 간접 참조관계를 나타내는 타겟요소 색인구조들을 노드들로 표현할 때 이들은 하나의 *타겟요소 색인계층(TEIH: Taget Element Index Hierarchy)*을 형성하고, $TEIH(T_0, E_1, T_1, E_2, \cdots, E_n, T_n)$ 또는 간단하게 $TEIH(0, n)$로 나타낸다. 여기서 가장 긴 경로의 색인노드 $TEI(0, n)$는 계층의 루트이다. $j-i > 1$인 각 노드 $TEI(i, j)$는 $0 < k, 1 < j-i$인 두 개의 자식 노드 $TEI(i, j-k)$와 $TEI(i+1, j)$를 가진다. $i = 0, 1, \ldots, n-1$에 대해서 색인노드 $TEI(i, i+1)$들은 계층의 바닥을 이루며, 이들은 B-TEI들이다. 특히, 이런 B-TEI들만 존재하는 색인계층을 우리는 *기본 타겟요소 색인계층(B-TEIH: Base - Taget Element Index Hierarchy)*이라 한다.

다음의 그림 4는 그림 3의 경로 스키마에 대한 두 가지 형태의 타겟요소 색인계층을 나타낸다. 그림 4(a)는 타겟요소 색인계층의 모든 색인노드가 존재

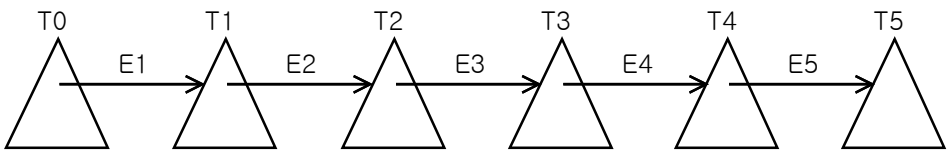
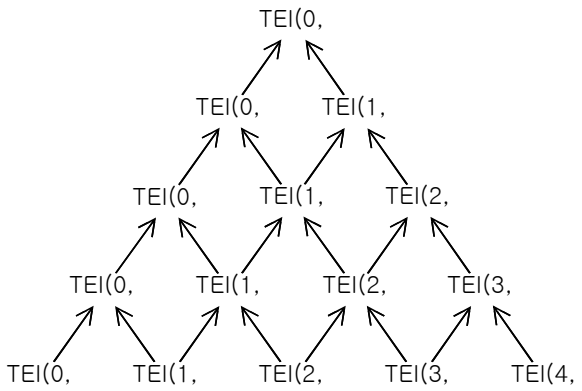
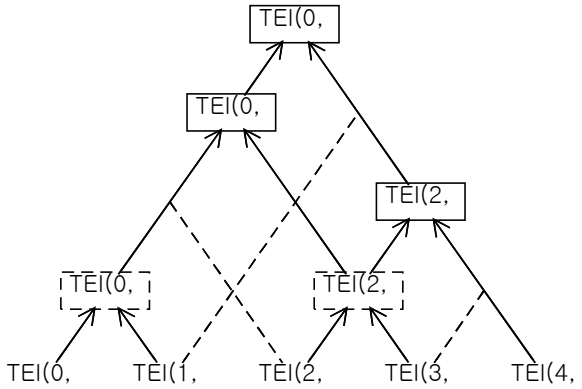


그림 3. 길이가 5인 XML 경로 스키마.



(a) 완전 타겟요소 색인계층(F-TEIH)



(b) 부분 타겟요소 색인계층(P-TEIH)

그림 4. 두 가지 형태의 타겟요소 색인계층.

하는 *완전 타겟요소 색인계층(F-TEIH: Full - Taget Element Index Hierarchy)*을 나타내고, 그림 4(b)는 타겟요소 색인계층의 일부 노드만 존재하는 *부분 타겟요소 색인계층(P-TEIH: Partial - Taget Element Index Hierarchy)*을 나타낸다.

그림 4(a)의 F-TEIH은 B-TEI들과 이들로부터 유도된 모든 길이의 경로에 대한 TEI들로 구성되어 있다. 이것은 경로 스키마를 따라 직접 또는 간접적으로 연결된 임의의 두 타입 사이의 관계를 하나의 색인구조만을 탐색하여 검색할 수 있다. 그림 4(b)의 P-TEIH은 F-TEIH에서 일부의 유도된 타겟요소 색인구조들만으로 구성된다. 이것은 오직 미리 주어진 두 타입들 사이의 타겟요소 색인구조들과 그들과 관련된 보조(유도된) 타겟요소 색인구조들만 생성함으로써, 미리 주어진 두 타입들 사이의 요소 쌍들에 대한 직접 검색만을 지원한다.

타겟요소 색인계층 *TEIH(0, n)*에서 기본 타겟요소 색인구조의 색인노드 *TEI(i, i+1) (i = 0, …, n-1)*들은 단계 1에 있으며, 루트 노드 *TEI(0, n)*은 단계 n에 있게 된다. F-TEIH은 크기가 매우 클 수 있고, 이런 경우 너무나 많은 색인구조의 구축이 있게 된다. 그러나 많은 경우에 색인계층의 모든 색인구조들을 구축할 필요는 없으며, 단지 자주 사용되는 탐색만을 지원하기 위한 색인구조들의 구축만이 필요하다. 즉 스키마 경로에서 자주 접근되는 부경로들이 주어졌을 때, 이들의 직접 탐색만을 지원하기 위한 P-TEIH을 구성하는 것이 바람직하다.

타겟요소 색인계층에서 자주 탐색되기 때문에 반드시 구축하여야 하는 색인노드들의 집합을 *목표 타겟요소 색인구조*라 한다. 예를 들어 그림 4(b)에서 *TEI(0, 5), TEI(0, 4), TEI(2, 5)*들(그림에서 실선 네모로 나타냄)이다. 그리고 주로 목표 타겟요소 색인구조의 갱신을 위해 생성해 둘 필요가 있는 그림 4(b)의 *TEI(0, 2), TEI(2, 4)*(그림에서 점선 네모로 나타냄)와 같은 색인노드들을 *보조 타겟요소 색인구조*라 한다. P-TEIH에서 목표, 보조 그리고 기본 색인노드들은 생성된 타겟요소 색인구조들이고, 나머지 색인노드들은 생성되지 않는 가상의 타겟요소 색인구조들이다.

목표 타겟요소 색인구조들의 갱신은 다음과 같은 세 종류의 데이터베이스 갱신에 따른 B-TEI들의 갱신으로부터 야기된다.

삽입: 타입 T_i의 요소 e_i에 타입 T_{i+1}의 요소 e_{i+1}를 삽입하는 경우로서, 이것은 *TEI(i, i+1)*에 색인 레코드 $(Eid(e_i), Tid(e_i), Eid(e_{i+1}), 1)$을 삽입하는 것에 해당한다.

삭제: 타입 T_i의 요소 e_i에 있는 타입 T_{i+1}의 요소 e_{i+1}를 삭제하는 경우로서, 이것은 *TEI(i, i+1)*로부터 색인 레코드 $(Eid(e_i), Tid(e_i), Eid(e_{i+1}), 1)$을 삭제하는 것에 해당한다.

수정: 타입 T_i의 요소 e_i에 있는 타입 T_{i+1}의 요소 e_{i+1}를 e'_{i+1}로 수정하는 경우로서, 이것은 *TEI(i, i+1)*에서 색인 레코드 $(Eid(e_i), Tid(e_i), Eid(e_{i+1}), 1)$을 삭제하고, 색인 레코드 $(Eid(e_i), Tid(e_i), Eid(e'_{i+1}), 1)$을 삽입하는 것에 해당한다.

본 논문에서는 하위단계의 두 타겟요소 색인구조를 병합하여 하나의 상위단계의 타겟요소 색인구조를 생성하거나 갱신하는 병합 연산자 "$\bowtie_m$"을 다음과 같이 정의한다. 즉 $TEI(i, k) \bowtie_m TEI(k, j)$는

*TEI(i, k)*의 색인 레코드 *(Eid(e_i), Tid(e_i), Eid(e_k), m_1)*와 *TEI(k, j)*의 색인 레코드 *(Eid(e_k), Tid(e_k), Eid(e_j), m_2)*가 병합되어 *TEI(i, j)*에 색인 레코드 *(Eid(e_i), Tid(e_i), Eid(e_j), $m_1 \times m_2$)*를 생성하게 된다. 즉, e_i에서 e_k까지는 m_1개의 서로 다른 요소 경로가 있고, e_k에서 e_j까지는 m_2개의 서로 다른 요소 경로가 있으면, e_i에서 e_j까지는 $m_1 \times m_2$ 개의 요소 경로가 있게 된다[2].

4. 부분 타겟요소 색인계층의 구축 알고리즘

부분 타겟요소 색인계층(P-TEIH)의 구축은 다음과 같은 세 단계로 이루어진다. 첫째로 주어진 목표 타겟요소 색인구조들의 생성에 필요한 보조 타겟요소 색인구조들을 찾는다. 둘째로 가장 낮은 단계의 기본 타겟요소 색인구조들을 생성한다. 그리고 셋째로 기본 타겟요소 색인구조들을 바탕으로 윗 단계의 목표 타겟요소 색인구조들과 보조 타겟요소 색인구조들을 병합연산을 통하여 생성한다.

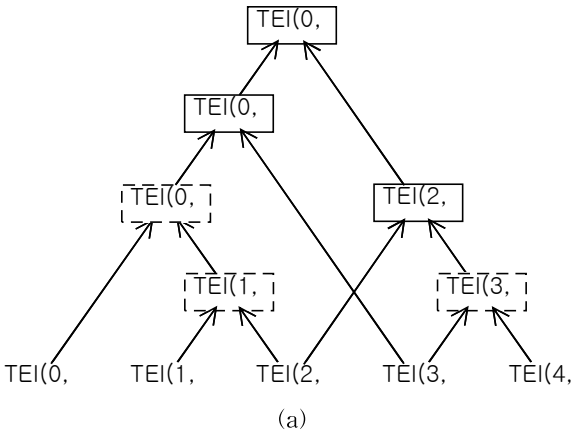
예제 1 그림 3의 경로 스키마에서 T_0와 T_5, T_0과 T_4, T_2와 T_5 사이의 관계탐색을 직접 지원하기 위한 목표 타겟요소 색인구조들의 집합은 *{TEI(0, 5), TEI(0, 4), TEI(2, 5)}*이고, B-TEI들의 집합은 *{TEI(0, 1), TEI(1, 2), TEI(2, 3), TEI(3, 4), TEI(4, 5)}*이다.

앞에서 나타낸 그림 4(b)는 이와 같은 목표 타겟요소 색인구조들에 대한 하나의 P-TEIH이고, 그림 5는 또 다른 두 종류의 P-TEIH을 나타낸다.

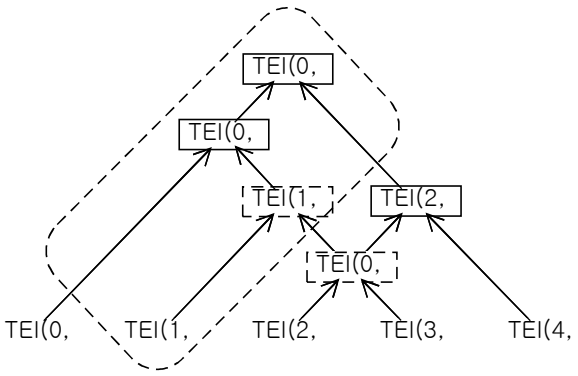
각 P-TEIH에서 보조 타겟요소 색인구조들로 그림 4(b)에는 {*TEI(0, 2), TEI(2, 4)*}, 그림 5(a)에는 {*TEI(0, 3), TEI(1, 3), TEI(2, 4)*}, 그리고 그림 5(b)에는 {*TEI(1, 4), TEI(2, 4)*}가 있다. □

주어진 목표 타겟요소 색인구조들에 대해서, 추가로 생성될 필요가 있는 색인구조들은 기본 및 보조 타겟요소 색인구조들이다. 이러한 보조 색인노드들의 집합은 하나 이상의 선택이 있을 수 있기 때문에, 최적의 선택은 보조 색인노드들의 총 개수와 목표 타겟요소 색인구조들의 생성 및 갱신을 위한 병합

2) n개의 동일한 색인 레코드*(Eid(e_i), Tid(e_i), Eid(e_j), m_k)(k = 1, 2, ..., n)*들은 이들의 경로의 수가 합산되어 $(EID(e_i), TID(e_i), EID(e_j), \sum_{k=1}^{n} m_k)$와 같은 하나의 색인 레코드로 합병된다.



(a)



(b)

그림 5. 목표 타겟요소 색인구조 {TEI(0, 5), TEI(0, 4), TEI(2, 5)}를 위한 두 종류의 P-TEIH

연산의 총 횟수를 최소화하는 것이어야 한다. 알고리즘 1은 이러한 최적의 보조 타겟요소 색인노드들을 선택하는 알고리즘이다.

알고리즘 1 보조 타겟요소 색인노드들의 선택

• 입력 정보
타입상속 계층 $T_0, \cdots, T_n$의 집합과 경로 스키마 $<T_0, E_1, T_1, E_2, \cdots, E_n, T_n>$에서 자주 참조되는 목표 타겟요소 색인노드들의 집합.

• 선택 방법
목표 색인노드의 집합을 생성하기 위하여 사용될 수 있는 보조 색인노드들의 집합들을 모아서, 다음과 같은 방법으로 최적의 보조 색인노드들의 집합을 선택한다.

단계1: 목표 색인노드들의 집합으로 시작하며, 그들의 직접 보조 색인노드들의 집합 *C*를 찾는다.

(1) 하나의 목표 색인노드 *TEI(i, j)*의 생성을 위한 직접 보조 색인노드들은 한 번의 병합 연산으로 *TEI(i, j)*를 생성할 수 있는 $i < k < j$인 k에 대한 두 개의 보조 색인노드 *{TEI(i, k), TEI(k, j)}* 쌍들의

집합이 된다. 여기에서 목표 또는 기본 색인노드가 되는 *TEI(i, k)* 또는 *TEI(k, j)*는 제거한다. 제거결과 두 노드 모두 삭제되는 k가 존재하면 *TEI(i, j)*를 위한 직접 보조 색인노드 집합은 공집합이 된다.

(2) 공집합이 아닌 목표 색인노드들은 하나의 직접 보조 색인노드 집합을 생성하고, 그 결과로 *C*는 집합들의 집합이 되어 *{{TEI(i, k), …, TEI(k, j)}, …, {TEI(i, m), …, TEI(m, j)}}*의 형태가 된다.

(3) *C*내의 집합 *s*의 각 색인노드에 대해 그것의 직접 보조 색인노드들을 찾는다. 만약 그것의 직접 보조 색인노드들이 *l*개의 집합, t_1, …, t_l로 구성된다면, *s*에 대한 *l*개의 복제본을 만들고, 각 복제본에 각 $t_i(1 < i < l)$를 더하여 새로운 *l*개의 집합을 만든다. 이 과정을 새로운 직접 보조 색인노드들이 발견되지 않을 때까지 반복한다. 그 결과는 목표 색인노드들의 집합을 생성하기 위해 사용되는 보조 색인노드 집합들의 집합이다.

단계2: *C*내의 각 집합 *s*에 대해 보조 색인노드들의 수가 최소가 되는 것들만 유지한다.

단계3: 단계 2에서 남아있는 노드들의 집합들로부터, 기본 색인노드의 갱신에 의한 목표 색인노드들과 보조 색인노드들을 갱신하는데 필요한 병합 연산의 횟수를 계산하고, 이 값이 최소가 되는 하나의 보조 색인노드 집합을 선택한다.

예제 2 에제 1에 대해서 알고리즘 1이 어떻게 동작하는지 알아본다. 처음 목표 색인노드들의 집합 *C = {{TEI(0, 5), TEI(0, 4), TEI(2, 5)}}*이다.

*TEI(0, 5)*를 생성하는데 사용될 수 있는 *TEI(0, 4)*는 목표 색인노드이고 *TEI(4, 5)*는 기본 색인노드이기 때문에, *TEI(0, 5)*의 직접 보조 색인노드 집합은 공집합이 된다. 그래서, *C = {{TEI(0, 4), TEI(2, 5)}}*이다.

목표 색인노드 *TEI(0, 4)*는 세 개의 직접 보조 색인노드 집합 *{TEI(0, 3)}*, *{TEI(0, 2), TEI(2, 4)}*, *{TEI(1, 4)}*를 갖고, 목표 색인노드 *TEI(2, 5)*는 두 개의 직접 보조 색인노드 집합 *{TEI(2, 4)}*와 *{TEI(3, 5)}*를 갖는다. 이 노드들 중에서 오직 *TEI(0, 3)*과 *TEI(1, 4)*만이 공집합이 아닌 보조 색인노드 집합을 가진다. 앞의 것은 *{TEI(0, 2)}*와 *{TEI(1, 3)}*을 갖고, 뒤에 것은 *{TEI(1, 3)}*과 *{TEI(2, 4)}*를 갖는다. 그러므로 가능한 보조 색인노드 집합들의 집합은 모든 이것들의 조합들이 되어야 한다. 즉,

*C = {{TEI(0, 3), TEI(0, 2), TEI(2, 4)}, {TEI(0, 3), TEI(0, 2), TEI(3, 5)}, {TEI(0, 3), TEI(1, 3), TEI(2, 4)}, {TEI(0, 3), TEI(1, 3), TEI(3, 5)}, {TEI(0, 2), TEI(2, 4)}, {TEI(0, 2), TEI(2, 4), TEI(3, 5)}, {TEI(1, 4), TEI(1, 3), TEI(2, 4)}, {TEI(1, 4), TEI(1, 3), TEI(3, 5)}, {TEI(1, 4), TEI(2, 4)}, {TEI(1, 4), TEI(2, 4), TEI(3, 5)}}*이다.

이들중 *{TEI(0, 2), TEI(2, 4)}*와 *{TEI(1, 4), TEI(2, 4)}*는 두 개의 노드를 가짐으로서 최소 수의 노드 집합들이다. 처음 것은 그림 4(b)에 있는 P-TEIH에 해당하고, 두 번째 것은 그림 5(b)에 있는 P-TEIH에 해당한다. 그림 4(b)와 그림 5(b)에서 기본 색인노드의 갱신에 의한 목표 색인노드들과 보조 색인노드들을 갱신하는데 필요한 병합 연산의 평균 횟수는 각각 3.2와 3.4이다. 이는 색인계층에서 각 기본 색인노드별로 이에 연결된 상위의 보조 및 목표 색인노드들의 개수를 세어서 평균값을 구한 것이다. 따라서 처음의 노드 집합 *{TEI(0, 2), TEI(2, 4)}*을 최적의 보조 타겟요소 색인구조들로 선택한다. □

P-TEIH에서는 생성된 색인노드에 해당하는 경로상의 두 타입들 사이의 요소 탐색을 하나의 색인구조의 검색으로 가능하게 할뿐만 아니라, 가상의 노드(생성되지 않은 색인노드)에 해당하는 두 타입 사이의 요소 탐색도 존재하는 두 개의 색인노드의 검색으로 대체가 가능하게 된다. 이는 그림 4(b)에서 어떠한 가상의 노드도 색인계층에 존재하는 두 개의 다른 노드들을 한 번의 병합 연산으로 구성할 수 있다는 사실로부터 알 수 있다.

한편으로 P-TEIH에서 하나의 타입 또는 다른 타입과의 관계에 대한 데이터베이스가 갱신되면 이에 따른 *TEI(k, k+1)*과 같은 기본 타겟요소 색인구조를 갱신하여야 한다. 그리고 이와 같은 기본 타겟요소 색인구조들의 갱신은 상위 단계에 있는 다른 타겟요소 색인구조들의 갱신에도 영향을 미치게 된다. 즉 임의의 기본 색인노드 *TEI(k, k+1)*의 갱신에 대해서 $i \leq k$이고 $j > k$인 모든 색인노드 *TEI(i, j)*들을 연이어서 갱신하여야 한다. 이와 같이 색인계층의 하위단계의 색인구조를 갱신하면 상위단계의 색인구조로 갱신이 전파되어 최상위의 목표 색인구조까지 갱신이 이루어진다. 예를 들어, 그림 5(b)에서 기본 색인노드 *TEI(1, 2)*가 갱신되면, 점선 내에 있는 영역의 노드 *{TEI(0, 4), TEI(1, 5), TEI(0, 5)}*들을 갱신하여야 한다.

5. 성능평가

본 절에서는 타겟요소 색인계층의 성능분석을 위한 색인구조로 제 2.3절에서 소개한 그림 2의 다차원 경로 색인구조(MD-PI)[17]와 함께 성능비용 모델을 제시하고 이를 바탕으로 성능평가 결과를 제시한다. 여기에서는 타겟요소 색인계층의 저장 공간의 크기, 그리고 질의 검색의 비용 등과 같은 성능평가 항목에 중점을 둔다. XML 경로 스키마 $<T_0, E_1, T_1, E_2, \cdots, E_n, T_n>$에 대해서 집합 $\{T_0, T_1, \ldots, T_n\}$는 타입상속 계층들을 나타낸다.

5.1 색인구조 비용 모델의 파라미터

다음 표 1은 성능평가 모델에서 사용한 데이터베이스 관련 파라미터들이고, 표 2는 색인구조에 관한 파리미터들이다.

타입상속 계층 T_i에서 요소 Ei의 구성요소 값으로 동일한 값을 가지는 요소의 평균 개수를 나타내는 매개변수 k(i)($0 \leq i \leq n$)는 구성요소의 공유정도를 나타내며, 이것은 질의처리 비용에 가장 크게 영향을 미친다. 앞으로 연속된 구성요소 공유도의 곱으로 다음과 같은 표기법을 사용한다.

$$k(i, j) = k(i) \times \cdots \times k(j) \quad (2)$$

그리고, 비용 모델을 간단히 하기 위해서 다음과 같은 몇 가지 가정을 한다.

- 타입 T_i의 각 요소는 타입 T_{i-1}의 요소들에 의해 반드시 참조된다. 이것은 매개변수의 관점에서는 D(i-1) = N(i)을 의미한다.
- 구성요소의 값들은 요소를 정의하는 타입의 요소들 중에서 균일하게 분포된다.

표 1. 데이터베이스 파라미터

파라미터	의 미
D(i)	타입상속 계층 T_i에서 요소 E_i의 구성요소가 가지는 유일한 값의 개수 ($1 \leq i \leq n$)
N(i)	타입상속 계층 T_i타입의 요소 개수($0 \leq i < n$)
k(i)	타입상속 계층 T_i에서 요소 E_i의 구성요소 값으로 동일한 값을 가지는 요소의 평균 개수($k(i) = [N(i)/D(i)]$)
EIDL	요소 식별자의 길이(8 바이트)
TIDL	타입 식별자의 길이(2 바이트)

표 2. 색인구조 파라미터

파라미터	의 미
P	페이지의 크기(4096 바이트)
pp	페이지 포인터의 길이(4바이트)
f	디렉토리 페이지의 평균 자식 노드수 ($0.69 * P/ (EIDL + TIDL + pp)$)
kl	키 값 필드의 길이(4바이트)
tidl	타입식별자(Tid) 필드의 길이(4바이트)
rl	레코드 길이 필드의 길이(2바이트)
ofl	오버플로 페이지 필드의 길이(4바이트)
neidl	*Eid* 또는 경로의 개수 필드의 길이(2바이트)
ol	*rl + tidl + ofl + neidl*(12바이트)
L	다차원 색인구조의 디렉토리 레코드의 길이($kl + tidl + pp$)
XB(i)	*i*번째 기본 타겟요소 색인구조의 평균 색인 레코드의 길이
XT	색인계층의 목표 타겟요소 색인구조의 평균 색인 레코드의 길이
XP	다차원 경로 색인구조의 평균 색인 레코드의 길이
NP	색인 페이지의 개수
NDP	디렉토리 페이지의 개수

- 모든 색인구조는 클러스터링 색인구조가 아니다. 이것은 요소들이 단말노드 색인 레코드들에 저장되어 있는 Eid들의 순서에 따라서 저장되지 않음을 의미한다.

본 논문에서는 각종 파라미터의 값으로 보편적인 색인구조 구성에서 사용하는 값들을 사용한다. 이들 값들은 파라미터 정의 부분에서 나타내었다.

5.2 저장 비용 평가

5.2.1 비용 모델

먼저 각 색인구조의 저장을 위한 비용 모델을 개발하고 이에 따라 이들을 비교한다.

(1) B-TEIH: 기본 타겟요소 색인계층을 구성하는 기본 색인구조의 경우, 먼저 n번째 색인구조의 크기를 $S(n)$이라 하면, 이는 n번째 색인구조의 색인 페이지의 개수 $NP(n)$과 디렉토리 페이지의 개수 $NDP(n)$를 합한 값이며 이들은 다음과 같이 계산된다.

$$XB(n) = k(n) * EIDL + kl + ol \tag{3}$$

$$NP(n) = \begin{cases} \lceil D(n) / \lfloor P/XB(n) \rfloor \rceil & \text{if } XB(n) \le P \\ D(n) * \lceil XB(n)/P \rceil & \text{if } XB(n) > P \end{cases} \tag{4}$$

색인 디렉토리 페이지의 개수는 다음식과 같이 나타낼 수 있다. 여기서 $LO = min(D(n), NP(n))$라 둔다.

$$NDP(n) = \lceil LO/f \rceil + \lceil \lceil LO/f \rceil /f \rceil + \ldots + X \tag{5}$$

이 식에서 각 인자는 마지막 인자 X가 f보다 작을 때까지 f값으로 계속해서 나누어진다. 마지막 인자 X가 1이 아니게 되면 전체에서 루트 노드를 위한 1이 더해진다.

NDP식을 나타내는 각 인자의 개수는 색인구조를 검색할 때 액세스되는 디렉토리 트리의 높이가 된다. 우리는 앞으로 이를 h로 나타낸다.

그리고 $i(0 \le i < n)$번째 기본 색인구조의 크기를 $S(i)$라 하면, 이는 i번째 색인구조의 색인 페이지의 개수 NP(i)와 색인 디렉토리 페이지의 개수 $NDP(i)$를 합한 값이며 이들은 다음과 같이 계산된다. 여기서 i번째 기본 색인구조의 키 값은 요소 식별자인 Eid이다.

$$XB(i) = (k(i)+1) * EIDL + ol \tag{6}$$

$$NP(i) = \begin{cases} \lceil D(i) / \lfloor P/XB(i) \rfloor \rceil & \text{if } XB(i) \le P,\ 0 \le i < n \\ D(i) * \lceil XB(i)/P \rceil & \text{if } XB(i) > P,\ 0 \le i < n \end{cases} \tag{7}$$

그리고 i번째 기본 색인구조의 디렉토리 페이지의 개수는 n번째의 경우와 같은 식(5)와 같이 나타낼 수 있다.

그러므로, B-TEIH 색인계층의 전체 크기는 모든 기본 색인구조들의 크기 $S(i)$ $(0 \le i \le n)$들의 합이 된다.

(2) P-TEIH, F-TEIH: 부분 또는 완전 타겟요소 색인계층을 구성하는 하나의 유도 색인구조 $TEI(i, j)$에 대한 색인구조의 크기를 $S(i, j)$라 하면, 색인 페이지의 개수는 다음과 같은 식으로 나타낼 수 있다. 먼저, 타입상속 계층 T_i에서 타입상속 계층 T_j의 동일한 E_j 요소를 참조할 요소의 개수는 식(2)의 $k(i, j)$와 같다.

$$XT = (k(i,\ j) + 1) * EIDL + ol \tag{8}$$

$$NP = \begin{cases} \lceil D(j) / \lfloor P/XT \rfloor \rceil & \text{if } XT \le P \\ D(j) * \lceil XT/P \rceil & \text{if } XT > P \end{cases} \tag{9}$$

그리고 유도 색인구조 $TEI(i, j)$의 디렉토리 페이지의 개수는 기본 색인구조의 경우와 같은 식(5)와 같이 나타낼 수 있다.

그러므로, P-TEIH, F-TEIH의 경우 색인계층의 전체 크기는 B-TEIH 색인계층의 색인구조들의 크기 $S(i)$ $(0 \le i \le n)$들의 합과 추가로 구축되는 모든 목표 색인구조들과 보조 색인구조들의 크기 $S(i, j)$들의 합이 된다.

(3) MD-PI: 경로상의 모든 경로 인스턴스를 색인 레코드에 가지고 있는 다차원 경로 색인구조의 경우 한 개의 색인 레코드에 가지고 있는 경로의 개수는 식(2)에서 $i = 0$이고 $j = n$인 $k(0, n)$개이다. 따라서 MD-PI 색인구조의 크기 S는 다음식의 색인 페이지의 개수 NP와 디렉토리 페이지의 개수 NDP의 합이 된다.

$$XP = k(0,\ n) * EIDL * n + kl + ol \tag{10}$$

$$NP = \begin{cases} \lceil D(n) / \lfloor P/XP \rfloor \rceil & \text{if } XP \le P \\ D(n) * \lceil XP/P \rceil & \text{if } XP > P \end{cases} \tag{11}$$

그리고 MD-PI 디렉토리 페이지의 개수 NDP는 다른 색인구조에서와 같은 방법으로 계산할 수 있다.

5.2.2 비교 평가

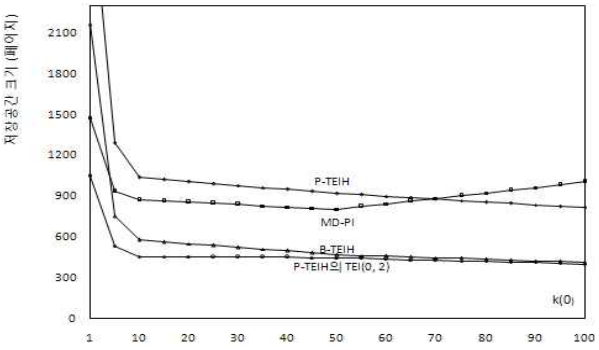
앞의 저장비용 모델에서 나타낸 비용식을 사용하여 각 색인구조의 저장비용을 계산할 수 있다. 우리는 평가를 단순히 하기 위하여 경로의 길이가 3인

경우에 한하여 B-TEIH. P-TEIH(경로의 길이가 3이면 F-TEIH과 같음), 그리고 참고문헌[17]에서 제시한 다차원 경로 색인구조 MD-PI와 같은 세 가지 형태의 색인구조의 크기를 비교한다. 경로의 길이가 3인 경우 B-TEIH는 두 개의 기본 색인구조 TEI(0, 1), TEI(1, 2)로 구성되며, 여기에 하나의 목표 색인구조 TEI(0, 2)를 추가하여 P-TEIH를 구성한다. 본 평가의 목적은 기본적으로 요소 공유도 k(i)에 대한 저장비용의 변화 정도를 각 색인구조별로 가름하기 위해서이다. 요소들 사이의 참조 정도를 정하기 위하여 N(0)를 200,000로 고정하고 D(0)과 D(1)을 변화시킴으로서 k(0)는 {1, 5, 10, 50, 100}의 값들이 되고 k(1)은 {1, 5, 10, 50}의 값들이 되도록 하였다.

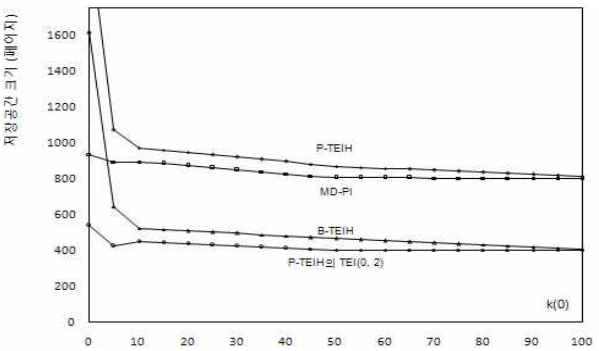
저장 공간에 대해서 경로를 저장하는 MD-PI의 경우 기본 색인구조들에 비해 많은 정도의 중복성을 가지게 된다. 경로 표현에서 경로를 구성하는 목표 요소들이 많은 참조를 공유하기 때문에, 이들의 경로 인스턴스에서 뒷부분이 같게 되는 많은 중복이 발생하게 되는 것이다. 이런 중복의 정도가 파라미타 k(i)에 의해서 표현되는 것이다. 그림 6(a)와 (b)는 각각 k(1)이 1과 5인 경우 k(0) 값의 변화에 대해 필요로 하는 세 가지 색인구조의 저장 공간 요구량을 나타낸다. k(1) 값이 10과 50인 경우에도 비슷한 관계를 나타낸다. 그림 6에서 T(0)의 요소들 사이의 참조 공유가 없는 경우, MD-PI가 B-TEIH에 비해 더 적은 저장 공간의 오버헤드를 가진다. 그러나 참조 공유도가 증가함에 따라, MD-PI에는 같은 정보가 여러 번 중복됨으로 인하여 B-TEIH에 비해 저장 공간의 오버헤드가 증가하게 된다. 그리고 참조 공유도가 증가할 때 B-TEIH와 P-TEIH의 목표 색인구조 TEI(0, 2)는 거의 같은 저장 공간의 오버헤드를 가진다. 이는 가정 1에 의해 타입 T(0)의 참조 공유도는 타입 T(1)의 요소 개수를 줄이기 때문이다. 여기서 F-TEIH의 전체 크기는 B-TEIH의 크기에 목표 색인구조 TEI(0, 2)의 크기를 더한 것과 같다. 그리고 여러 개의 색인구조들로 구성된 P-TEIH의 경우에도 전체 크기가 하나의 다차원 경로 색인구조인 MD-PI의 크기에 비해 약간 상회하는 크기를 가짐을 알 수 있다.

5.2 검색 비용 평가

본 절에서는 세 가지 색인구조 B-TEIH와 P-TEIH(또는 F-TEIH)의 목표 타겟요소 색인구조



(a) k(1) = 1일 때



(b) k(1) = 5일 때

그림 6. 경로의 길이가 3인 경우 B-TEIH, P-TEIH, MD-PI, 그리고 P-TEIH의 목표 색인구조 TEI(0, 2)의 저장 공간 비용

TEI(0, n), 그리고 MD-PI에 대해서 타겟요소 탐색을 위한 색인구조의 검색비용을 비교검토 한다.

5.2.1 비용 모델

TEI(0, n)과 MD-PI의 경우 타겟요소 탐색을 위해서 오직 한 번의 색인구조 검색이 필요하지만, B-TEIH의 경우 n개의 기본 색인구조를 차례대로 검색하여야 한다. 다음은 각 색인구조별로 타겟요소 탐색을 위한 색인구조의 검색에 필요한 페이지 접근 횟수 A를 계산하는 식을 나타낸다.

(1) TEI(0, n)과 (2) MD-PI의 경우: 색인 레코드의 크기를 X라 두면, 색인 페이지 접근 회수 A는 다음식과 같다.

$$A = \begin{cases} h+1 & \text{if } X \le P \\ h+\lceil X/P \rceil & \text{if } X > P \end{cases} \quad (12)$$

여기서 TEI(0, n)의 경우 X = XT이고, MD-PI의 경우 X = XP이다.

(3) B-TEIH의 경우: 먼저 n번째 기본 색인구조에 대한 검색비용을 A(n)으로 나타내고, 그 다음에 i(0

≤ i < n)번째 기본 색인구조에 대한 검색비용을 A(i)로 나타낸다.

n번째 색인구조의 색인 페이지 접근 회수 A(n)은 다음식과 같다.

$$A(n)=\begin{cases} h(n)+1 & \text{if } X(n)\le P \\ h(n)+\lceil XB(n)/P\rceil & \text{if } X(n)>P \end{cases} \quad (13)$$

*i*번째 색인구조에 접근하는 색인 페이지의 수를 결정하기 위해서는 먼저 *(i+1)*번째 색인구조에서 검색된 *Eid*의 개수를 정하여야 한다. 이를 *NEID(i)*로 나타내면 다음과 같다.

$NEID(n-1) = k(n)$이고,

$$NEID(i) = k(n) * k(n-1) * \cdots * k(i+1),\quad 0 \le i < n-1 \quad (14)$$

주어진 *NEID(i)*개의 Eid에 대해서 접근해야 될 색인 페이지의 개수를 계산하기 위하여 우리는 Yao [20]의 공식를 사용한다. 이 공식은 m개의 페이지에 분배된 n개의 레코드에서 임의의 k개의 레코드를 탐색하기 위한 페이지 접근의 횟수를 나타낸다.

$$y(k, m, n)=\lceil m*(1-\prod_{i=1}^{k}\frac{n-n/m-i+1}{n-i+1})\rceil \quad (15)$$

이 공식을 사용함으로서 *i*번째 기본 색인구조를 검색하기 위해 접근해야할 색인 페이지의 개수 *AL(i)*를 얻을 수 있다.

$$AL(i)=\begin{cases} y(\neq \mathrm{ID}(i), LP(i), D(i)) & \text{if } XB(i)\le P \\ \neq \mathrm{ID}(i) * \lceil XB(i)/P\rceil & \text{if } XB(i)>P \end{cases} \quad (16)$$

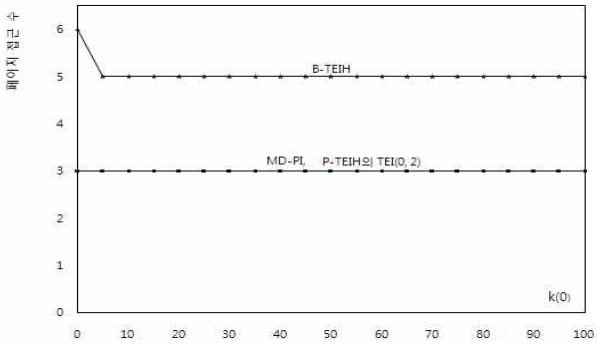
따라서 여기에 색인 디렉토리 트리의 높이 *h(i)*를 더하면 i번째 기본 색인구조에 대한 검색비용 *A(i)*가 된다.

$$A(i)=\begin{cases} y(\mathrm{NEID}(i), LP(i), D(i))+h(i) & \text{if } XB(i)\le P \\ \mathrm{NEID}(i) * \lceil XB(i)/P\rceil + h(i) & \text{if } XB(i)>P \end{cases} \quad (17)$$

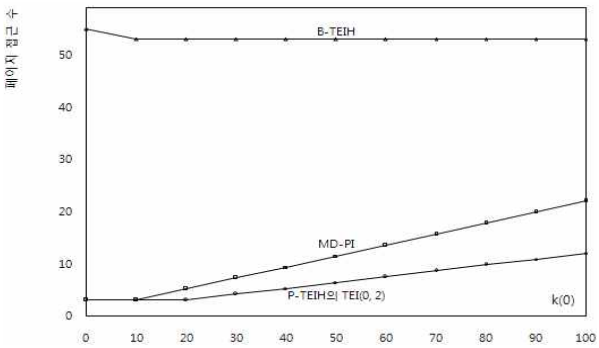
5.2.2 비교 평가

앞의 검색비용 모델에서 나타낸 비용식을 사용하여 각 색인구조의 검색비용을 계산할 수 있다. 우리는 평가를 단순히 하기 위하여 먼저 경로의 길이가 3인 경우에 한하여 검색비용을 검토한다. 각 파라미터들에 대한 값들은 저장비용 모델에서 사용한 값과 같은 값을 사용한다.

우리는 저장비용에서처럼 공유도 k(i)에 대한 검색비용을 비교한다. 그림 7(a)와 (b)는 k(1) 값이 1과 50일 때, k(0) 값의 변화에 따른 색인구조들의 검색비용을 보이고 있다. 타입 T(1)의 참조 공유도가 낮을 때(k(1) = 1)는 B-TEIH의 페이지 접근수가 나머지 두 색인구조에 대해서 그리 많지 않지만, k(1)의 값이 크게 되면, 그 차이도 매우 크게 된다. 이는 두 번째 기본 색인구조로부터 탐색된 타입 T(1)의 요소 식별자 Eid의 개수가 k(1)에 의해서 주어지기 때문이다. 이 Eid 각각에 대해서 이 요소를 참조하는 타입 T(0)의 요소 식별자 Eid를 검색하기 위해서 첫 번째 색인구조가 매번 탐색되어야 하기 때문이다. 따라서 참조 공유도 k(i)는 B-TEIH에서 검색비용에 매우 큰 영향을 미치게 된다.



(a) 참조 공유도 k(1) = 1일 때



(b) 참조 공유도 k(1) = 50일 때

그림 7. 경로의 길이가 3인 경우 요소 탐색을 위한 각 색인구조의 검색 비용

색인계층의 목표 색인구조인 TEI(0, 2)의 경우와 MD-PI의 경우에는 색인 레코드의 크기가 페이지 크기보다 크지 않는 한 k(0)와 k(1) 값의 변화에도 같은 검색비용을 나타낸다. 색인 레코드의 크기가 페이지의 크기를 초과한 후에는 두 색인구조의 검색비용은 차이가 나게 된다. 그러므로 그림 7(b)에서 k(0)의 값이 10에서 20에서는 목표 색인구조 TEI(0, 2)의 페

이지 접근수가 일정하게 유지 되지만, MD-PI의 페이지 접근수는 k(1)의 값이 10일 때부터 증가함을 볼 수 있다. 이는 MD-PI 색인 레코드의 크기가 TEI(0, 2)의 색인 레코드의 크기보다 훨씬 빠르게 증가하기 때문이다.

평가결과를 요약하면, B-TEIH의 경우 검색비용은 k(1)의 값에 매우 크게 영향을 받는다. 즉, k(1)의 값에 비례해서 페이지 접근수가 증가한다. 반면에 나머지 두 색인구조의 경우에는 k(0) * k(1)의 값이 특정 값보다 크게 될 때 이 값에 영향을 많이 받게 되지만, 그렇지 않으면 일정한 검색비용을 갖게 된다. 이 특정 값은 TEI(0, 2) 색인구조인 경우에는 1000이고, MD-PI 색인구조인 경우에는 500이다. k(0) * k(1)의 값이 이 특정 값보다 크게 되면 색인구조의 접근비용이 증가하지만, 그 증가속도는 B-TEIH에서의 경우 k(1)에 비례하여 증가하는 경우보다는 훨씬 작게 된다.

이와 같은 특정 값은 다음과 같이 정해진다. TEI(0, 2)에서 k(0) * k(1) $\leq$ *500* 이면 저장비용 모델에서 색인 레코드의 크기 XT의 계산식(8)에서 색인 레코드의 크기가 페이지의 크기보다 작게 된다. 그러므로 단지 하나의 단말 색인 페이지만 접근하면 된다. 그리고 500 < k(0) * k(1) $\leq$ *1000* 인 경우에는 색인 레코드가 두 개의 색인 페이지를 점유하게 되어 두 번의 색인 페이지 접근이 필요하다. 하지만 이 경우에는 색인구조의 디렉토리 트리의 높이가 1 감소하게 되어, 단말 색인 페이지 접근의 증가를 상쇄하게 된다. 그러나 k(0) * k(1) > *1000* 이 되면, 색인 레코드는 두 개 이상의 색인 페이지를 점유하게 되어 디렉토리 트리의 높이 감소가 이를 상쇄하지 못하게 되고, 검색비용이 증가하게 된다. MD-PI에서도 k(0) * k(1) > *250* 인 경우에 색인 레코드의 크기가 페이지 크기보다 크게 되어 TEI(0, 2)에서와 같은 설명이 가능하다.

6. 결 론

본 논문에서는 XML 데이터베이스의 조상-자손 관계 탐색을 지원하기 위한 다차원 타입상속 색인구조(MD-TIX)를 효율적으로 구축하기 위하여 타겟요소 색인계층 접근법을 제안하였다. 타겟요소 색인계층은 직접 또는 간접 요소관계 탐색을 위한 타겟요소 색인구조들의 집합으로 계층을 이룬다. 본 논문에서는 세 종류의 타겟요소 색인계층인 기본 타겟요소 색인계층(B-TEIH), 완전 타겟요소 색인계층(F-TEIH), 그리고 부분 타겟요소 색인계층(P-TEIH)을 제시하였다. 특히 가장 이상적인 P-TEIH 색인계층의 구축을 위해서 갱신 전파를 최소로 하는 보조 타겟요소 색인구조들의 선택 알고리즘을 제안하였다.

B-TEIH에는 기본 타겟요소 색인구조들만 존재하므로 타입상속 계층 T_{i+l}에서 T_i로의 탐색은 *l*개의 연속적인 기본 타겟요소 색인구조(*TEI(i+l-1, i+1), …, TEI(i, i+1)*)들의 검색을 필요로 하므로 검색 성능이 매우 좋지 않다. 반면에 B-TEIH를 기반으로 몇 개의 목표 타겟요소 색인구조들을 단계적으로 추가하여 구축하는 P-TEIH의 경우에는 하나 또는 두 개의 타겟요소 색인구조만의 검색을 필요로 하므로 검색 성능을 크게 향상시킬 수 있다.

F-TEIH에서는 모든 타겟요소 색인구조들을 생성하기 때문에, F-TEIH의 구성에서 알고리즘 1과 같은 보조 타겟요소 색인노드들의 선택 알고리즘의 수행이 필요 없다. 즉, 각 단계의 모든 타겟요소 색인구조들이 목표 타겟요소 색인구조들로 간주된다. 만약 경로 운행이 P-TEIH에서 직접 생성되지 않은 가상의 노드에 해당하는 검색이 필요하다면, F-TEIH의 사용이 P-TEIH의 사용과 비교해서 보다 빠를 수 있다. 그러나 F-TEIH는 P-TEIH보다 너무 많은 저장 공간과 갱신 비용을 필요로 한다.

따라서 본 논문에서는 XML 스키마 경로의 길이가 4이상인 경우 구축과 관리가 어려웠던 다차원 타입상속 색인기법에서 보조 타겟요소 색인구조들의 선택 알고리즘을 이용한 색인계층 접근법으로 P-TEIH를 구축함으로써 질의처리에 효율적으로 이용할 수 있음을 보였다.

참 고 문 헌

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

[2] C. W. Chung, J.K. Min, and K. Shim. “APEX: An Adaptive Path Index for XML Data,” *Proc. Intl. Conf. on Management of Data, ACM SIGMOD*, pp. 121—132, 2005.

[3] S.C. Haw and C.S. Lee, “Extending Path Sum-

mary and Region Encoding for Efficient Structural Query Processing in Native XML Databases," *The Journal of Systems and Software*, Vol. 82, No. 6, pp. 1025-1035, 2009.

[4] J.H. Lee, "An Assignment Method of Multidimensional Type Inheritance Indexes for XML Query Processing," *Journal of Korea Multimedia Society*, Vol. 12, No. 1, pp. 1-15, 2009.

[5] 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.

[6] 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.

[7] B.F. Cooper et al., "A Fast Index for Semistructured Data," *Proc. Intl. Conf. on Very Large Data Bases*, pp. 341- 350, 2001.

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

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

[10] A. Kemper and G. Moerkotte, "Access Support Relations: An Indexing Method for Object Bases," *Information Systems*, Vol. 17, No. 2, pp. 117-145, 1992.

[11] R. Goldman and J. Widom, "DataGuides: Enable Query Formulation and Optimization in Semistructured DataBases," *Proc. Int'l Conf. on Very Large Data Bases*, pp. 436-445, 2003.

[12] K.P. Leela, and J.R. Haritsa, "Schema-conscious XML indexing," *Information Systems*, Vol. 32, No. 2, pp. 344-364, 2007.

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

[14] T. Chen, J. Lu, and T.W. Ling, "On Boosting Holism in XML Twig Pattern Matching Using Structural Indexing Techniques," *Proc. of the 2005 ACM SIGMOD international conference on Management of data*, pp. 455-466, 2005.

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

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

[17] 이종학, "MD-TIX: XML 질의의 효율적 처리를 위한 다차원 타입상속 색인기법," 멀티미디어학회논문지, 제10권, 제9호, pp. 1093-1105, 2007.

[18] K.C. Kim et al., "Acyclic Query Processing in Object-Oriented Databases," *Proc. Intl. Conf. on Entity-Relationship Approach*, pp. 329-346, 1989.

[19] 이종학, "2D-THI: XML 데이테베이스를 위한 이차원 타입상속 계층색인," 멀티미디어학회논문지, 제9권, 제3호, pp. 265-278, 2006.

[20] S. B. Yao, "Approximating Block Accesses in Database Organizations," *Communications of the ACM*, Vol. 20, No. 4, pp. 260-261, 1977.

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 : 색인구조, 다차원 파일구조, 데이터베이스 설계, XML 데이터베이스, 데이터 웨어하우스, 생물정보학, 데이터베이스 보안 등