# 준구조적 데이타베이스에서의 정규경로표현 최적화를 위한 질의전지 기법

## (A Query Pruning Technique for Optimizing Regular Path Expressions in Semistructured Databases)

박 창 원 † 정 진 완 ††
(Chang-Won Park) (Chin-Wan Chung)

**요 약** 정규경로표현은 전통적 스키마를 가정하지 않는 준구조적 데이타에 대해 질의를 고안하기 위한 기본적 질의 요소이다. 그리고 질의전지는 정규경로표현의 처리에 있어 불필요한 탐색을 제거하기 위한 중요한 최적화 기법이다. 그러나 기존 질의전지는 다중 정규경로표현들은 완전히 최적화하지 못하는 경우가 많으며, 기존 질의전지의 결과를 후처리하는 기존의 방법은 지수적으로 증가하는 많은 부분결과들의 조합들을 확인해야 한다. 본 논문에서는 전처리 단계와 전지 단계로 구성된 두 단계 질의전지라 부르는 새로운 기법을 소개한다. 우리의 두 단계 질의전지는 다중 정규경로표현의 최적화에 효과적이며, 지수적으로 증가하는 많은 부분결과들의 조합들을 확인하지 않는다는 점에서 기존의 방법보다 더 확장성이 있다.

**키워드** : 준구조적 데이타, 질의전지, 정규경로표현, 최적화

***Abstract*** Regular path expressions are primary elements for formulating queries over the semistructured data that does not assume the conventional schemas. In addition, the query pruning is an important optimization technique to avoid useless traversals in evaluating regular path expressions. However, the existing query pruning often fails to fully optimize multiple regular path expressions, and the previous methods that post-process the result of the existing query pruning must check exponential combinations of sub-results. In this paper, we present a new query pruning technique that consists of the preprocessing phase and the pruning phase. Our two-phase query pruning is effective in optimizing multiple regular path expressions, and is more scalable than the previous methods in that it never check the exponential combinations of sub-results.

**Key words** : Semistructured Data, Query Pruning, Regular Path Expression, Optimization

## 1. 서 론

전통적 스키마를 가정하지 않는 준구조적 (semi structured) 데이타[1,2,3]는 그 구조가 불규칙적이거나, 일부만 알려져 있거나, 자주 변경되는 데이타이다. 그러므로 정확한 구조 지정 없이 질의를 하기 위해 정규경로표현이 매우 중요하다. 그러나, 정규경로표현의 처리는 필요 이상으로 많은 에지(edge)를 지나야 하기 때문에 불필요한 에지는 지나지 않도록 하는 것이 중요한 최적화의 하나이다. 이러한 측면에서, 정규경로표현을 포함한 질의를 준구조적 그래프 스키마[2, 4, 5, 6]를 이용해서 좀 더 최적화된 질의로 재작성 하는 질의전지(質疑剪枝; query pruning)[7, 8] 기법이 개발되었다.

하지만 기존의 질의전지는 다중 경로표현들을 완전히 최적화하지 못하는 경우가 많다. 이는 정규경로표현들 사이의 상관관계를 무시하는 경우가 많기 때문이며, 질의의 답에 기여하지 않는 스키마 수준 초과 경로들을 야기한다. 더 나쁜 점은 단 하나의 스키마 수준 초과 경로라도 데이타 수준에서 질의를 처리하는 동안에는 훨씬 더 많은 경로들을 지나야하는 결과를 초래할 수 있다는 것이다. 비록 기존 질의전지의 결과를 개선하기 위한 후처리가 가능하지만, 이는 정규경로표현의 수에 대해 지

· 본 연구는 교육인적자원부의 Brain Korea 사업에 의해 지원받았음.
† 비 회 원 : LG전자기술원 정보기술연구소 선임연구원
cwpark@islab.kaist.ac.kr
†† 종신회원 : 한국과학기술원 전산학과 교수
chungcw@islab.kaist.ac.kr
논문접수 : 2001년 2월 16일
심사완료 : 2002년 3월 12일

수적으로 증가하는 모든 정규경로 표현의 부분결과들의 조합을 확인해야만 한다[7]. 따라서 이전의 후처리 방법은 정규경로표현의 수에 대해 지수적으로 증가하는 시간이 걸리며, 기존 질의전지에서 사용되지 않았다[7].

그럼에도 불구하고 다중 경로표현들은 분기(branching) 경로표현[10]과 같은 복잡한 질의에 매우 빈번하게 사용된다. 또한 복잡한 질의는 그 처리부담이 크므로 효과적 최적화 기법이 매우 긴요하다. 본 논문은 기존 질의전지의 시간 복잡도를 그대로 유지하면서도 다중 경로표현의 최적화에 효과적인, 전처리 단계와 전지 단계로 구성된 새로운 두 단계 질의전지 기법을 소개한다. 우리의 두 단계 질의전지는 다중 정규경로표현의 최적화에 효과적이며, 지수적으로 증가하는 많은 부분결과들의 조합들을 확인하지 않는다는 점에서 기존 질의전지와 후처리의 경우보다 더 확장성이 있다. 본 논문에서 우리는 스키마가 커지고, 정규경로표현의 수가 증가해도 안정된 성능을 보인다는 측면에서 효율성보다는 확장성이라는 용어를 사용한다.

본 논문의 구성은 다음과 같다. 2절에서는 관련연구를 다루고, 3절에서 논문 전체에 걸쳐 사용될 예제를 다룬다. 4절에서는 기존 질의전지와 그 문제점을 설명하고, 그 해결을 위한 새로운 개념을 5절에서 정의한다. 6절에서는 두 단계 질의전지를 전처리 단계와 전지 단계로 나누어 기술하고, 두 단계 질의전지의 정확성 및 유효성을 보인다. 마지막으로 7절은 본 논문에서 제시된 기법이 기존의 기법보다 좋은 점을 실험결과를 통해 보이며, 8절에서 논문 전체의 결론을 맺는다.

## 2. 관련연구

XSQL[11]에서는 확장된 경로표현을 도입하여 속성변수나 경로 변수 등을 이용해서 객체지향 스키마에 대한 검색은 물론, 임의의 경로를 통한 데이타 검색을 지원하고 있다. 그리고 일반화된 경로표현[12]은 속성 변수나 경로 변수를 이용한 경로표현으로 스키마의 일부를 고의로 무시함으로써 문서에 대한 다양한 단위의 질의를 지원하고 있다. 이상의 두 가지 경로표현은 정규경로표현과 매우 흡사하지만 미리 정의된 전통적 스키마를 가정하고 있으며, 보조적 질의 요소이다. 이에 반하여, 정규경로표현은 그러한 가정을 배제한 준구조적 데이타를 위한 기본적 질의 요소이다[1, 2, 3, 13, 14].

질의전지[7, 8]는 준구조적 데이타에서 정규경로표현을 최적화하기 위한 대표적인 기법이다. 이는 기존의 뷰(view)를 이용한 질의 재작성[15]과는 다르며, 준구조적 데이타를 위한 뷰를 이용한 정규경로표현의 재작성[7, 9]도 활발히 연구되고 있다. 또 다른 최적화 방법으로 준구조적 데이타를 위한 비용기반(cost-based) 최적화 기법[10, 16]에 확장된 경로표현과 일반화된 경로표현을 위한 비용기반 최적화기법[17]을 적용하는 방법도 생각할 수 있으나, 이는 고정된 전통적 스키마를 데이타와 동일하게 취급하도록 비용기반 최적화 기법을 확장하는 것이므로, 고정된 전통적 스키마를 가정하지 않는 준구조적 데이타에는 적합하지 않다. 이 가운데 [7]에서 제시된 질의전지는 본 논문에서 제시하는 두 단계 질의전지의 기반이 되므로 4절에서 자세히 설명한다.

## 3. 예제

**데이타.** 그림 1에 예제 데이타를 보였다. 따옴표 없는 문자열로 표시된 라벨들은 구조 성분으로 생각할 수 있다. 값들은 단말노드에만 있으며, 따옴표 있는 문자열로 표시되어 있다. 최상단의 Research_organizations 에지는 일반 에지가 아니라 데이타의 루트를 가리키는 광역변수이다.

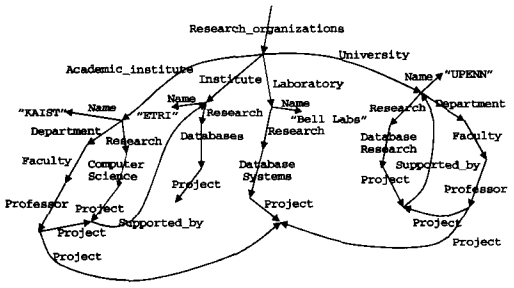


그림 1 준구조적 데이타의 예제

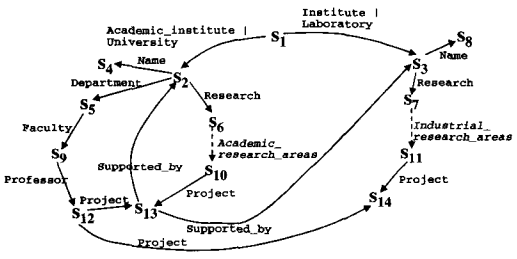


그림 2 준구조적 그래프 스키마의 예제

**스키마.** 그림 2는 예제 데이타가 따르는 예제 스키마이다. 스키마는 데이타와 비슷하지만 각 에지는 "|"로 구분되는 가능한 모든 라벨들의 집합으로 표시될 수 있고 단말노드라도 값이 올 수 없다는 것이 다르다. 구조

의 간결한 표현을 위해 복잡한 일부 구조를 묶어 단순화하였다. 한 예로, 점선으로 표시된 *Academic_research_areas* 에지는 에지들의 집합을 나타내며, 그 에지들 각각은 각 학술 연구 조직의 특정 연구 분야를 나타낸다. 그리고 모든 스키마 노드 s마다 s로 분류되는 데이타 노드들의 집합으로 외연(extent)을 정의하며, 이를 ext(s)라 표시한다. 예를 들어, ext($s_4$)는 {"KAIST," "UPENN"}이다

**질의.** [7]의 질의언어를 사용하여 질의를 작성하였는데, [7]과는 달리 임의의 단일 에지와 에지열을 표현하기 위해 각각 "_"과 "*"를 사용하였다.

```
select C
where *.Professor.Project.A in Research_organizations,
                                                    (조건 1)
      _.B in Research_organizations,                (조건 2)
      Name.C in B,                                  (조건 3)
      Research._.Project.A in B,                    (조건 4)
      Supported_by.B in A                           (조건 5)
```

조건 R.$v_2$ in $v_1$은 $v_1$이 가질 수 있는 데이타 노드에서 $v_2$가 가질 수 있는 데이타 노드로 경로표현 R에 부합하는 경로가 있으면 그 값이 참이다. 질의의 where절에 있는 모든 조건들을 만족시키는 변수들에 대한 데이타 노드들의 대입을 $\theta$라 하자. 그러면 위 질의의 의미는 모든 $\theta$를 찾아 변수 C가 가질 수 있는 모든 데이타 노드들을 구하는 것이며, 교수가 참여하고 있는 프로젝트를 스스로 지원하고 있는 연구 조직의 이름들을 검색한다. 예제 데이타에 대한 처리 결과는 "UPENN"이다.

**변수의 역할.** 예제 질의의 조건 5에서 A와 B는 각기 기점(source)변수와 종점(destination)변수이다. 조건 1과 5에서 A는 조건 1의 종점변수이고 조건 5의 기점변수이므로 매개(intermediate)변수라 한다. A를 거친 조건 1에서 5로의 이러한 상호작용은 입출력 상호작용이라고 한다. 그리고 임의의 입출력 상호작용들의 연속으로 순서를 부여할 수 있는 조건들을 정렬가능 조건들이라고 부른다. 조건 1과 5는 정렬가능 조건들의 예이다. 조건 3과 4에서는 B가 두 조건의 기점변수이므로 분기(branch)변수라 한다. 조건 1과 4에서는 A가 두 조건의 종점변수이므로 접합(join)변수라 한다. 매개변수, 분기변수, 그리고 접합변수는 모든 대입 $\theta$에서 그 변수가 각 조건에서 유일한 데이타 노드를 가지게 하기 위한 것이다. 마지막으로, 변수 C는 질의에 의해 검색되므로 결과변수이다.

## 4. 기존 질의전지 및 문제점

이 절은 본 논문에서 제시하는 두 단계 질의전지의 기반이 되는 기존 질의전지를 예제를 이용해 자세히 설명하며, 기존 질의전지가 가진 문제점과 그 보완방법도 함께 설명한다. 기존 질의전지는 주어진 다중 조건들을 가지고 다음 단계들을 순서대로 진행한다.

**프로덕트 오토마타 (product automata) 구성.** 예를 들어 조건 2의 "_"를 이용한 프로덕트 오토마타[7]의 구성을 고려하면 다음과 같다: (1) 그림 3에 도시한 "_"에 해당하는 비결정적 오토마타 $A_2$를 구성한다; (2) 그림 2의 스키마 S와 $A_2$에 대해서 $14\times2$개의 상태들 ($s_1$,$a_1$), ($s_1$,$a_2$), …, ($s_{14}$,$a_2$)로 이루어진 프로덕트 오토마타 $S\times A_2$를 구성한다. 전이는 S에 $s\text{—}l\rightarrow s'$ 에지가 있고 $A_2$에 $a\text{—}l'\rightarrow a'$ 전이가 있을 때 l'가 l과 같거나 와일드카드이면 $(s,a)\text{—}l\rightarrow(s',a')$ 전이이다. 초기상태는 s가 S에서 부합확인의 시작점이고 a가 $A_2$의 초기상태인 (s,a) 상태들이다. 조건 2의 경우에는 S의 루트인 $s_1$이 항상 부합확인의 시작점이므로 ($s_1$,$a_1$)이 초기상태이다. 일반적으로, 부합확인의 시작점은 고정되지 않고 초기상태도 고정되지 않는다. 최종상태는 a가 $A_2$의 최종상태인 (s,a) 상태들이다. 남아있는 다른 조건들과 S의 나머지 프로덕트 오토마타도 동일하게 구성된다. 그 다음 단계는 AND/ OR 그래프를 이용하여 그림 4에 도시된 전지된 프로덕트 오토마타 $S\sqcap A_2$를 만든다. $S\sqcap A_2$는 초기상태에서 최종상태로의 모든 유효한 경로 위에 있는 $S\times A_2$의 상태들과 전이들로 구성된다.

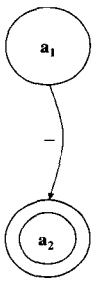


그림 3 오토마타 A2

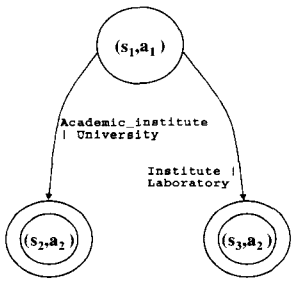


그림 4 전지된 프로덕트 오토마타 $S\sqcap A_2$

**AND/OR 그래프 구성.** AND/OR 그래프[7]에서 OR 노드는 이전 노드들 일부가 접근가능하면 접근가능한 노드인 반면, AND 노드는 이전 노드들 모두가 접근가능해야만 접근가능한 노드이다. 우선, 프로덕트 오토마타 $S\times A_k$ 모두의 합집합을 취하되, 그들의 상태들을 모두 OR 노드로 한다. 두 번째로, 조건들 사이의 묵시

적 상호작용들을 다음과 같이 나타낸다: (1) 어떤 조건 R.z *in* x를 단독으로 고려할 때, 최소한 해당 조건의 오토마타 A의 어떤 최종상태 a에 대해서 (s,a)가 전지된 프로덕트 오토마타 S⊓A 내에 존재하면 그리고 오직 그럴 때만 z는 ext(s)의 어떤 노드를 가질 수 있다. 이는 OR 조건이며, S의 각 스키마 노드 s와 조건 R.z in x의 프로덕트 오토마타 S×A마다 하나의 OR 노드 (s,z,A)를 만들어 나타낸다. 더불어 S×A의 모든 최종상태 (s,a)마다 (s,a)→(s,z,A) 에지들을 만든다. (2) 하나의 변수 z에 대한 모든 $R_1.z$ in $x_1$, ⋯, $R_n.z$ in $x_n$ 조건들을 고려할 때, z가 이 조건들 하나 하나에서 ext(s)의 어떤 노드를 가질 수 있으면 그리고 오직 그럴 때만 z는 ext(s)의 어떤 노드를 가질 수 있다. 이는 AND 조건이며, 모든 변수 z와 S의 모든 스키마 노드 s마다 하나의 AND 노드 (s,z)를 만들어 나타낸다. 조건 $R_i.z$ *in* $x_i$마다 $(s,z,A_i)$→(s,z) 에지들을 만든다. (3) 각 $R_1.z$ in x, $R_2.y$ *in* z 쌍의 매개변수 z를 거친 입출력 상호작용들을 $R_2.y$ in z의 프로덕트 오토마타 S×A′마다 (s,z)→(s,a) 에지들을 만들어 나타낸다. 여기서 a는 A′의 초기상태이다.

마지막으로, AND/OR 그래프의 최대 접근가능성(accessibility property)[7], 즉 접근가능한 노드들의 최대 집합을 구한다. S의 루트 노드 s와 광역변수 v를 위한 AND 노드 (s,v)는 계산하는 동안 항상 접근가능하다고 간주한다. 접근가능한 초기 OR 노드로부터 접근가능한 최종 OR 노드까지의 모든 경로 상의 접근가능한 OR 노드들과 전이들로 전지된 프로덕트 오토마타 S⊓$A_k$를 정의한다. S×$A_k$의 다른 모든 노드들은 접근불가능하다고 간주한다. 그림 5는 접근가능한 노드들만 포함하는 결과 AND/OR 그래프를 간략하게 도시한 것이다.

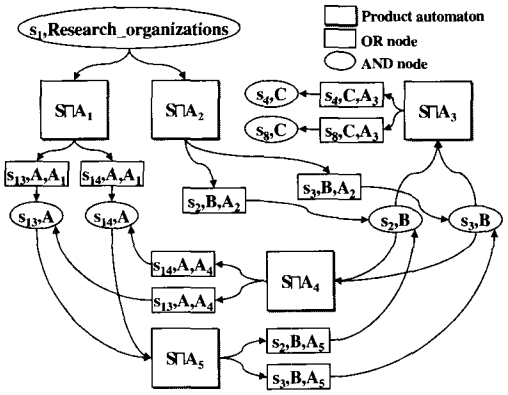


그림 5 결과 AND/OR 그래프

**결과 최적 질의 작성.** 모든 접근가능한 AND 노드 (s,z)는 변수 z가 ext(s)에 속하는 데이타 노드들을 가질 수 있다는 것을 의미한다. 질의전지는 각 전지된 프로덕트 오토마타의 접근가능한 어떤 초기 OR 노드부터 접근가능한 어떤 최종 OR 노드까지의 모든 경로들을 이용하여 결과 최적 질의를 작성한다. 예를 들어, 그림 5에서 AND 노드 ($s_1$,Research_organizations)가 접근가능하므로, 그림 4의 S⊓$A_2$에서 접근가능한 초기 노드 ($s_1$,$a_1$)부터 접근가능한 최종 노드들 ($s_2$,$a_2$)와 ($s_3$,$a_2$)까지의 경로들을 이용해서 "_" 대신에 "(Academic_institute | University | Institute | Laboratory)"를 얻는다. 이 결과 표현은 그림 2의 S에서 $s_1$부터 $s_2$와 $s_3$까지의 "_"에 부합하는 경로들을 관찰하여 확인할 수 있다.

질의전지는 정규경로표현들의 상태들의 수와 스키마의 스키마 노드들의 수에 대하여 PTIME(polynomial time)에 작업을 마치며[7], 예제 질의의 결과 최적 질의는 다음과 같다

```
select C
where (Academic_institute | University).Professor.Project.
A in Research_organizations,                          (조건 1)
      (Academic_institute | University | Institute |
       Laboratory).B in Research_organizations,        (조건 2)
      Name.C in B,                                    (조건 3)
      Research.(Academic_research_areas | Industrial_
            research_areas).Project.A in B,           (조건 4)
      Supported_by.B in A                             (조건 5)
```

이상이 기존 질의전지에 포함된 단계들이다. 지금부터는 위 단계들로 이루어진 기존 질의전지의 문제점과 그 보완방법을 설명한다.

**문제점.** 위의 결과 질의는 완전히 최적화되지 않았음을 유의하기 바란다. Supported_by 에지들이 없는 부-그래프(sub-graph)들은 지날 필요가 없다. 하지만, 조건 2에 의해 Institute와 Laboratory의 부-그래프를 초과로 탐색해야한다. 그리고, 조건 4에는 Institute와 Laboratory만을 위한 연구 분야들(Industrial_research_areas)이 나타난다. 이러한 완전히 최적화되지 않은 조건들에 의해 스키마 수준 초과 경로들이 발생한다. 예를 들어, 그림 5의 AND/OR 그래프는 ($s_2$,B)와 ($s_3$,B)로부터 조건 4(Research._.Project.A in B)의 S⊓$A_4$를 거쳐 ($s_{13}$,A)와 ($s_{14}$,A)가 접근가능함을 나타내고 있다. 하지만, 조건 4와 그림 2의 스키마에 따르면 $s_2$로부터는 $s_{13}$만 도달할 수 있고, $s_{14}$는 도달할 수 없다. 이러한 질의전지의 문제는 어떤 정렬가능 조건들 내의 각 조건마다 전술한 상관관계를 무시하는 것에 기인한다.

**보완 방법.** 질의전지를 후처리로 보완하는 이전의 방법들[7]은 입출력 상호작용에 의해 서로 연관된 각 조건 쌍마다 개개의 부분결과들의 조합을 해당 프로덕트 오토마타를 이용해서 확인해야만 한다. 예를 들어 그림 5에서는 광역변수 Research_organizations는 $s_1$에 해당하는 데이타 노드를 가질 수 있고 $S \sqcap A_2$를 통하여 변수 B는 $s_2$ 또는 $s_3$에 해당하는 데이타 노드를 가질 수 있다. 가장 간단한 예로 $S \sqcap A_2$에서 $S \sqcap A_3$으로의 입출력 상호작용만을 생각하면 변수 Research_organizations, B, C에 대하여 가능한 $2^2$개 조합들 각각이 유효한 조합인지를 일일이 $S \sqcap A_2$와 $S \sqcap A_3$을 이용해서 확인해야만 한다. 우선 <Research_organizations, B, C> 순서대로 <$s_1$, $s_2$, $s_4$> 조합의 경우를 보자. $S \sqcap A_2$(그림 4)에서 $S \sqcap A_2$의 초기상태 가운데 $s_1$과 곱해진 상태 ($s_1$,$a_1$)에서 최종상태 가운데 $s_2$와 곱해진 상태 ($s_2$,$a_2$)로 도달할 수 있는 경로가 있는지 확인하고, 동시에 $S \sqcap A_3$에서 $s_2$와 곱해진 초기상태에서 $s_4$와 곱해진 최종상태로 도달할 수 있는 경로가 있는지 확인해야 한다. 이 부분결과 조합은 모든 경로가 존재하므로 유효한 조합이다. 반면에 <$s_1$, $s_2$, $s_8$> 조합의 경우 여전히 $S \sqcap A_2$에서는 $s_1$과 곱해진 초기상태에서 $s_2$와 곱해진 최종상태로의 경로가 존재하지만 $S \sqcap A_3$에서는 $s_2$와 곱해진 초기상태에서 $s_8$과 곱해진 최종상태로의 경로가 존재하지 않아 유효하지 않은 조합이다 (이는 그림 2의 스키마에서 $s_2$로부터 $s_8$로의 경로가 없는 것을 관찰하면 확인할 수 있다). 이러한 방법을 AND/OR 그래프 내에 존재하는 모든 입출력 상호작용 조건들에 적용해야 하므로 이전의 방법들은 정규경로표현의 수에 대해 지수적으로 증가하는 시간이 걸리며, 이와 같은 후처리는 채택되지 않았다[7].

## 5. 최소 최대화 조건 집합

최소 최대화 조건 집합은 앞 절에서 설명한 기존의 후처리를 이용한 보완 방법과는 달리 각각의 정렬가능 조건들의 집단마다 개별적 프로덕트 오토마타들 대신에 단일 프로덕트 오토마타를 만들어서 전술한 상관관계들을 보존할 수 있게 한다. 이러한 측면에서, 우리는 하나의 정렬가능 조건들의 집단에 속하는 조건들을 다른 하나의 더 긴 조건으로 서로 연결하기 위한 두 규칙들을 제시한다. 규칙에서 Ri와 vi는 각기 임의의 경로표현과 변수이다.

**규칙 1.** R1.v0 in v1, R2.v2 in v0≡R1.R2.v2 in v1

**규칙 2.** R1.v0 in v1, R2.v2 in v0≡R1.$^{v0}$R2.v2 in v1

규칙 1에서와 같이 경로표현 쌍을 서로 연결한 후 매개변수를 제거함에 있어서 주의가 필요하다. 이는 그 매개변수가 분기변수, 접합변수, 그리고 결과 변수로서의 다른 역할들을 하고 있을 수 있기 때문이다. 그런 경우, 규칙 2에서 ".v0"로 표시한 새로운 내부 구성체인 변수 주석(variable annotation)을 이용해서 그러한 정보를 보존한다.

두 규칙들은 어떤 조건 집합 내의 정렬가능 조건들로 이루어진 임의의 서로 겹치는 집단들에 적용될 수 있다. 겹치는 집단들이 원래의 조건 집합에 포함된 모든 조건들을 포함하고 있다면, 결과 조건 집합은 원래의 조건 집합과 동등하다. 이는 겹치는 집단들을 만족시키는 모든 대입 $\theta$가 원래의 조건 집합 또한 만족시키고 반대의 경우도 마찬가지이기 때문이다. 그리고 그런 각 집단에서 정렬가능 조건들이 어떤 일련의 입출력 상호작용들에 의해 순환(cycle)을 이룬다면 그 조건들은 임의의 반복된 순서로 서로 연결될 수 있다. 따라서, 그런 순환적(cyclic) 조건들로부터 무한히 많은 동등한 조건 집합들을 얻을 수 있다. 여기서 우리는 모든 가능한 조건 집합들 가운데 가장 바람직한 집합이 어떤 것인지 정의할 필요가 있다. 어떤 질의가 주어졌을 때, 그 질의의 조건들에서 기점변수로만 사용되는 변수를 시작변수라 하자. 반대로, 그 질의의 조건들에서 종점변수로만 사용되는 변수는 마침변수라 하자.

**정의 1.** (*최소 최대화 조건*) 최소 최대화 조건은 어떤 질의의 정렬가능 조건들을 규칙 1과 규칙 2를 이용하여 서로 연결해서 만들어지는 조건으로 다음을 만족한다.

- 그 조건의 기점변수는 시작변수이다;
- 그 조건의 종점변수를 제외한 모든 변수들이 서로 다르다;
- 그 조건의 종점변수는 그 조건에 포함된 모든 변수들이 서로 다르면 마침변수이고, 그렇지 않으면 동일 변수 쌍의 두 번째 동일 변수이다.

**정의 2.** (*최소 최대화 조건 집합*) 어떤 질의의 최소 최대화 조건 집합은 그 질의 내에 존재하는 모든 최소 최대화 조건을 포함하는 집합이다.

## 6. 두 단계 질의전지

두 단계 질의전지는 전처리 단계와 전지 단계로 구성된다. 질의가 주어지면 전처리 단계는 해당 질의의 최소 최대화 조건 집합을 구하고, 전지 단계는 최소 최대화 조건 집합을 최적화한다.

### 6.1. 전처리 단계

전처리 단계는 조건 집합을 상호작용 그래프로 구성

하고, 해당 그래프에 깊이 우선 탐색(depth first search; DFS)을 수행하여 모든 최소 최대화 조건을 찾는다.

**상호작용 그래프 구성.** (1) 각 조건의 기점변수 s와 종점변수 d를 두 노드로 하고, 경로표현 p를 s로부터 d로의 방향성 에지의 라벨로 만든다; (2) 같은 변수는 같은 노드가 되도록 한다. 구성 도중에 각 변수의 역할들을 결정하는 것을 돕기 위해 각 변수에 대한 추가 정보를 모은다. 결과변수의 경우, 그 사실을 해당 노드에 저장한다. 그리고 각 변수의 인입도(in-degree)와 인출도(out-degree)를 해당 노드에 저장한다. 그러한 정보를 이용해서 표 1에서와 같이 변수의 역할들을 바로 결정할 수 있다.

표 1 변수의 역할 결정

| 인입도 | 인출도 | 역할 |
|---|---|---|
| =0 | ≥1 | 시작변수 |
| ≥1 | ≥1 | 매개변수 |
| ≥0 | ≥2 | 분기변수 |
| ≥2 | ≥0 | 접합변수 |
| ≥1 | =0 | 마침변수 |

**DFS 기반 전처리 알고리즘.** 이 알고리즘은 DFS를 이용해서 상호작용 그래프 내의 각 시작변수로부터 어떤 마침변수로의 모든 단순경로를 찾는다. 여기서 단순경로는 (1) 순환이 없으면 모든 변수가 서로 다르고, (2) 순환이 있을 때는 한 변수가 두 번째 나타나면 그 변수에서 끝나는 경로를 말한다. (2)의 경우, 다른 변수들은 서로 다르다. 시작변수로부터 마침변수까지의 단순경로 하나는 최소 최대화 조건 하나에 해당한다.

알고리즘은 두 개의 부분 알고리즘으로 구성되며, 아래 알고리즘 1에 주어져 있다. 알고리즘 *Preprocess*는 주 알고리즘으로 각 시작변수마다 어떤 마침변수로의 모든 단순경로를 구하기 위해 알고리즘 *Interconnect*를 불러 DFS를 시작한다. 각 부-루틴의 의미는 그 이름이 의미하는 그대로이다. 특히, *Append*(∅, e)는 빈 경로표현의 끝에 하나의 경로표현 e를 추가함을 의미한다. *Annotate*(p, d)는 경로표현 p의 끝에 변수주석 d를 만든다.

**시간 복잡도.** 주어진 조건들이 n개라면 상호작용 그래프를 구성하기 위해서는 각 조건마다 n-1개의 다른 조건들을 확인해야 한다. 그러므로, 상호작용 그래프를 만들고 추가 정보를 저장하기 위한 시간 복잡도는 $O(n^2)$이다. 그리고 최소 최대화 조건 집합을 구하기 위

알고리즘 1 DFS 기반 전처리 알고리즘

```
Algorithm Preprocess
Input: 상호작용 그래프 G
Output: 최소 최대화 조건 집합
Begin
  clear visited flags for all variables;
  for each variable v in G do
    if v is a starting variable then
      for each edge e of v
            and its destination de do
        Interconnect(v, Append(∅, e), de);
      end for
    end if
  end for
End

Algorithm Interconnect
Input: 기점변수 s, 경로표현 p, 종점변수 d
Output: 최소 최대화 조건 (s, p, d)
Begin
  if d is a terminating variable or visited then
    Output(s, p, d);
  else
    Mark_visited(d);
    if d is a branch variable
        or a join variable
        or the result variable then
      Annotate(p, d);     // 규칙 2
    end if
    for each incident edge e
          and its destination de of d do
      Interconnect(s, Append(p, e), de);     // 규칙 1
    end for
    Clear_visited(d);
  end if
End
```

해서는 각 시작변수마다 O(n)의 DFS로 모든 방향성 에지를 탐색해야 한다. 서로 다른 변수들의 수는 n에 비례하므로 최소 최대화 조건 집합을 구하기 위한 시간 복잡도도 $O(n^2)$이다. 따라서 전체 시간 복잡도는 $O(n^2)$이다.

**예제.** 예제 질의의 상호작용 그래프는 그림 6에 도시하였다. 전처리 단계는 해당 그래프에서 모든 최소 최대화 조건을 찾으며, 아래와 같이 네 개의 최소 최대화 조건들을 찾게 된다:

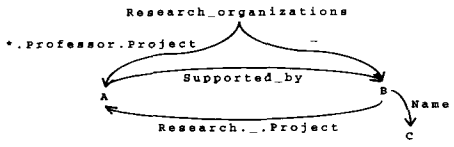


그림 6 예제 질의의 상호작용 그래프

(1) *.Professor.Project.$^{A}$Supported_by.$^{B}$Name.C in Research_organizations

(2) *.Professor.Project.$^{A}$Supported_by.$^{B}$Research._.Project.A in Research_organizations

(3) _.$^{B}$Research._.Project.$^{A}$Supported_by.B in Research_organizations

(4) _.$^{B}$Name.C in Research_organizations

### 6.2. 전지 단계

전지 단계는 기존 질의전지의 확장판이다. 핵심 메카니즘은 변경되지 않았으므로 가장 중요한 확장인 변수주석을 처리하기 위한 확장만을 제시한다.

**프로덕트 오토마타 구성에서의 확장.** 어떤 조건에서 하나의 점은 그 조건을 위한 비결정적 오토마타에서 하나의 상태에 해당한다. 따라서 한 점이 변수주석을 동반하면 해당 상태는 같은 변수로 주석이 달린다. 또한 오토마타 A의 상태 a가 변수 v로 주석이 달리면 프로덕트 오토마타 S×A의 해당 상태 (s,a)들도 v로 주석이 달린다. 변수주석 v를 표시하기 위해 $a^{v}$와 $(s,a^{v})$에서와 같이 위 첨자를 이용한다.

**AND/OR 그래프 구성에서의 확장.** 동일한 변수 v로 주석이 달린 모든 상태들 $(s_i,a_j^{v})$을 고려할 때, 그러한 상태들 각각에서 v가 $ext(s_i)$의 어떤 노드를 가질 수 있으면 그리고 오직 그럴 때만 v는 $ext(s_i)$의 어떤 노드를 가질 수 있다. 이 AND 조건은 AND/OR 그래프의 AND 노드 $(s_i,v)$를 사용해 처리된다. 그리고 상태 $(s_i,a_j^{v})$를 포함하는 프로덕트 오토마타 $S\times A_k$를 고려하자. AND/OR 그래프 내의 $(s_i,v)$가 접근가능할 때만 $S\times A_k$의 $(s_i,a_j^{v})$로부터의 모든 경로들이 계속될 수 있다. 이러한 AND 조건과 계속 조건을 처리하기 위해 전지 단계는 각 $(s_i,a_j^{v})$마다 하나의 AND 노드 $(s_i,a_j,v)$를 만들고, 그들을 다음과 같이 연결한다: (1) $(s_i,a_j^{v})$의 모든 전이들을 $(s_i,a_j,v)$로 옮긴다; (2) 하나의 $(s_i,a_j^{v})\rightarrow(s_i,a_j,v)$ 에지를 추가한다; (3) $(s_i,a_j,v)$와 AND 노드 $(s_i,v)$에 대하여 한 쌍의 에지 $(s_i,a_j,v)\rightarrow(s_i,v)$과 $(s_i,v)\rightarrow(s_i,a_j,v)$를 추가한다. 이 에지 쌍은 S의 $s_i$와 $A_k$의 $a_j^{v}$에 대해서 $S\times A_k$ 내의 $(s_i,a_j^{v})$는 유일하고 $(s_i,a_j,v)$도 유일하므로 직접 연결될 수 있다.

전지 단계는 이전과 같이 최대 접근가능성을 계산하고 전지된 프로덕트 오토마타를 만든다. $S\times A_k$ 내에서 $(s_i,a_j,v)$가 접근가능한 초기상태에서 어떤 최종상태로의 경로 상에 있지 않으면 $S\sqcap A_k$ 내에서 $(s_i,a_j,v)$는 접근불가능하게 된다. $(s_i,a_j,v)$가 접근불가능할 때 $(s_i,v)$도 또한 접근불가능하며, 반대의 경우도 마찬가지이다. 대조적으로 기존의 질의전지는 $(s_i,a_j^{v})$ 상태의 AND 조건과 계속 조건 모두를 고려하지 않는다. 대신, 종점변수가 v인 주어진 질의의 어떤 원래 조건을 위한 별개의 프로덕트 오토마타 $S\times A_l$와 v와 $S\times A_l$를 위한 OR 노드 $(s_i,v,A_l)$를 사용한다.

**결과 최적 질의 작성에서의 확장.** 모든 전지된 프로덕트 오토마타 $S\sqcap A_k$의 접근가능한 초기상태에서 접근가능한 최종상태로의 각 경로를, 그 경로 상의 접근가능한 AND 노드들 (s,a,v)로 구분된 각 부-경로와 동등한 조건들의 집합으로 바꾼다. 그리고 나서, 모든 중복된 조건들을 제거하고, 공통의 기점변수에서 공통의 종점변수로의 조건들을 "|"로 구분된 라벨들의 집합을 이용해서 병합한다.

**시간 복잡도.** 변수주석은 AND/OR 그래프의 모든 프로덕트 오토마타에서 제한된 수의 상태들을 위해서 사용된다. 따라서 변수주석들에 의해 도입되는 (s,a,v) 형태의 추가적 AND 노드들을 고려할 때, 기존 질의전지의 시간 복잡도는 AND 노드들의 총 수의 증가에 의해 악화되지 않는다. 결과적으로 전지 단계는 정규경로표현들의 상태들의 수와 스키마의 스키마 노드들의 수에 대하여 PTIME에 작업을 마친다.

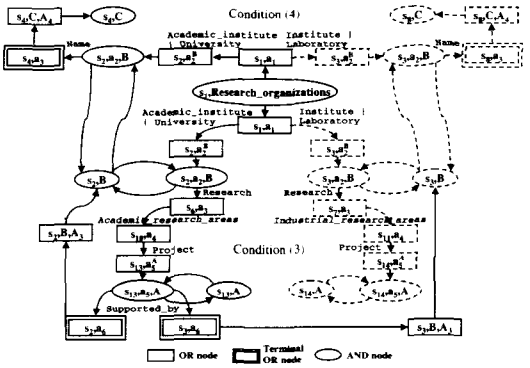


그림 7 예제 질의를 위한 결과 AND/OR 그래프의 일부

**예제.** 이전 부-절의 끝에 있는 예제 질의의 최소 최대화 조건들을 다시 고려하자. 그림 7에 조건 (3)과 조건 (4)의 전지된 프로덕트 오토마타와 AND/OR 그래프의 관련 부분들을 도시하였다. 점선으로 그려진 노드들은 접근불가능한 반면에 실선으로 그려진 노드들은 접근가능하다. 그림 5의 AND/OR 그래프와는 다르게, 조건 (3)의 $(s_3,a_2,B)$와 $(s_{14},a_5,A)$가 접근불가능하므로 $(s_3,B)$와 $(s_{14},A)$가 접근불가능하다. 그리고 $(s_3,B)$와 조건 (4)의 $(s_3,a_2,B)$가 접근불가능하기 때문에 $(s_8,C)$가 접근불가능하다. 따라서 결과 최적 질의는 다음과 같다:

```
select C
where (Academic_institute | University).Professor.
      Project.A  in Research_organizations,        (조건 1)
      (Academic_institute | University).B in Research_
       organizations,                              (조건 2)
      Name.C in B,                                 (조건 3)
      Research.Academic_research_areas.Project.A in B,
                                                   (조건 4)
      Supported_by.B in A                          (조건 5)
```

이 결과 질의는 완전히 최적화되었음을 유의하기 바란다. 라벨이 Institute와 Laboratory인 에지들을 만나면 그 부-그래프들은 결코 지나지 않는다. 그리고 조건 4에 Institute와 Laboratory만을 위한 연구 분야들(*Industrial_research_areas*)은 나타나지 않는다.

### 6.3. 정확성과 유효성

정확성(correctness)과 유효성(effectiveness)을 보이기 위해서 $\theta(v)$를 대입 $\theta$에서 변수 v에 대입되는 데이타 노드라 하자. 그러면 전지 단계의 정확성은 다음과 같이 표현될 수 있다. 스키마 S에 따르는 임의의 데이타 D와 Q의 임의의 변수 v에 대해서, S에 속하는 어떤 스키마 노드 si에 있어서 $\theta(v) \in ext(si)$인 Q의 변수들의 대입 $\theta$가 존재하면 G에서 AND 노드 (si,v)가 접근가능하다. 또 한편으로, 그 역은 전지 단계가 유효한 대입들만을 발견한다는 뜻으로 유효성을 의미한다.

**정리 1.** 임의의 질의 Q와 스키마 S에 있어서 QS를 S에 대해서 두 단계 질의전지가 생성한 최적화된 질의라 하자. QS는 S에 관해 Q와 동등하며, 스키마 수준 초과 경로를 포함하지 않는다.

**증명.** S에 따르는 임의의 데이타 D와 Q의 임의의 변수 v에 대해서, S에 속하는 어떤 스키마 노드 $s_i$에 있어서 $\theta(v) \in ext(s_i)$인 Q의 변수들의 대입 $\theta$가 존재하면 그리고 오직 그럴 때만 Q를 위한 AND/OR 그래프 G에서 AND 노드 $(s_i,v)$가 접근가능함을 보인다.

첫째로, 정확성은 $\theta$에 기반한 G의 접근가능성 $A$를 구성하여 보인다. $A$는 다음 노드들로 구성된다. (1) $\theta(v) \in ext(s_i)$인 모든 AND 노드 $(s_i,v)$; (2) $\theta(v) \in ext(s_i)$인 모든 OR 노드 $(s_i,v,A_k)$; (3) 조건 $R_k.v_{k2}$ in $v_{k1}$의 전지된 프로덕트 오토마타 $S \sqcap A_k$ 내의 모든 상태 $(s_i,a)$와 AND 노드 $(s_i,a,v)$. 이 때 $A_k$에 의해 승인되는 $\theta(v_{k1})$부터 $\theta(v_{k2})$까지의 경로가 D 내에 존재해서 해당 경로 상의 중간 노드 $x \in ext(s_i)$가 $A_k$의 상태 a에 대응되어야 한다. $\theta$가 Q의 모든 조건을 만족시키는 대입이기 때문에 A는 실로 접근가능성이다. 따라서 $\theta(v) \in ext(s_i)$면 $(s_i,v)$는 접근가능하다.

둘째로, 유효성은 모순을 유도하여 보인다. 어떠한 임의의 $\theta$라도 $\theta(v) \not\in ext(s_i)$라 가정하자. Q의 원래의 조건 $R_k.v$ in $v_s$와 그 조건의 오토마타 $A_k$를 고려하자. 가정에 의해 $A_k$은 $\theta(v_s)$부터 어떤 데이타 노드 $x \in ext(s_i)$로의 어떠한 경로도 승인하지 않는다. 전처리 이후에 해당 조건은 최소한 하나의 최소 최대화 조건에 병합된다. 따라서 그 최소 최대화 조건의 오토마타 $A_K$의 일부는 $A_k$에 해당된다. 바로 그 부분을 $A_K{:}A_k$이라고 표시하고 그 최종상태를 $a_k$이라고 표시하자. 프로덕트 오토마타 $S \times A_k$를 고려하자. $A_K$의 최종상태가 $A_K{:}A_k$의 최종생태와 같으면 $A_K{:}A_k$이 $(s_i,a_k)$로의 어떤 경로도 승인하지 않기 때문에 OR 노드 $(s_i,v,A_K)$과 AND 노드 $(s_i,v)$는 접근불가능하다. 그렇지 않으면 $(s_i,a_k)$을 지나는 모든 경로는 $A_K{:}A_k$에 의해서 접근불가능하게 된다. 만약 $A_K{:}A_k$의 $a_k$에 v로 주석이 달려 있으면 $A_K{:}A_k$이 $(s_i,a_k^v)$로의 어떤 경로도 승인하지 않기 때문에 AND 노드 $(s_i,a_k,v)$와 $(s_i,v)$는 접근불가능하다. 이러한 결과는 $(s_i,v)$가 접근가능하다는 충분조건에 모순이 된다. 따라서 $(s_i,v)$가 접근가능하면 $\theta(v) \in ext(s_i)$이다.

## 7. 실 험

우리는 새로운 두 단계 질의전지가 효과적이면서도 확장성이 있다는 것을 실험적으로 보이기 위해 Java SDK 1.3.1을 사용하여 원형 시스템을 구현하였다. 그리고 전형적인 3 가지 스키마와 각 스키마에 대한 질의를 만들어 RAM 192MB를 장착한 Pentium III 700MHz PC에서 실험을 수행하였다.

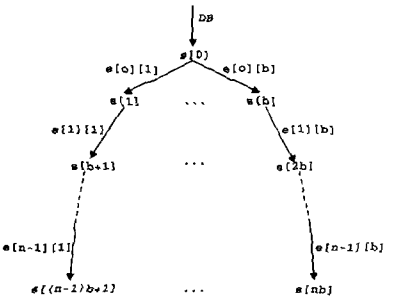


그림 8 선형 스키마

**스키마 및 질의.** 선형 스키마, 분지형 스키마, 순환형/접합형 스키마는 전형적인 스키마들이다. 선형 스키마는 그림 8에 도시되어 있으며, 광역변수인 DB가 가리키는 s[0] 노드에서의 분지의 수를 의미하는 b 값과 깊이

표 2 실험에 사용한 질의

| 종류 | 질의명 | 스키마 깊이 | 정규경로표현들 |
|---|---|---|---|
| 선형 | Ql2 | 2 | _.X1 in DB, e[1][1].X2 in X1 |
| | Ql3 | 3 | _.X1 in DB, _.X2 in X1, e[2][1].X3 in X2 |
| | Ql4 | 4 | _.X1 in DB, _.X2 in X1, _.X3 in X2, e[3][1].X4 in X3 |
| | Ql5 | 5 | _.X1 in DB, _.X2 in X1, _.X3 in X2, _.X4 in X3, e[4][1].X5 in X4 |
| 종류 | 질의명 | 분지 깊이 | 정규경로표현들 |
| 분지형 | Qb0 | 0 | _.X1 in DB, _.X12 in X1, e[1][1][1].X13 in X12,<br>_.X2 in DB, _.X22 in X2, e[15][15][15].X23 in X22 |
| | Qb1 | 1 | _.X1 in DB, _.X12 in X1, e[1][1][1].X13 in X12,<br>_.X22 in X1, e[1][15][15].X23 in X22 |
| | Qb2 | 2 | _.X1 in DB, _.X12 in X1, e[1][1][1].X13 in X12,<br>e[1][1][15].X23 in X12 |
| 종류 | 질의명 | 접합 깊이 | 정규경로표현들 |
| 순환형 또는 접합형 | Qj1 | 1 | _.X1 in DB, _.X2 in X1, _.X3 in X2, _.X4 in X3, e[4][1].X1 in X4 |
| | Qj2 | 2 | _.X1 in DB, _.X2 in X1, _.X3 in X2, _.X4 in X3, e[4][1].X2 in X4 |
| | Qj3 | 3 | _.X1 in DB, _.X2 in X1, _.X3 in X2, _.X4 in X3, e[4][1].X3 in X4 |
| | Qj4 | 4 | _.X1 in DB, _.X2 in X1, _.X3 in X2, _.X4 in X3, e[4][1].X4 in X4 |

를 의미하는 n 값을 변화시키면서 실험하였다. 깊이는 광역변수인 DB가 가리키는 s[0] 노드의 깊이를 0으로 하여 한 단계 깊어짐에 따라 차례로 1씩 증가한다. b 값은 5, 10, 15의 순서로 증가시켰고, 각 b 값에 대해서 n 값은 2, 3, 4, 5의 순서로 증가시켰다. 그리고 n의 값에 따라 선형 질의 내의 정규경로표현의 수를 2, 3, 4, 5의 순서로 증가시켰다. 실험에 사용한 각 질의의 조건들은 표 2에 나열하였다.

그림 9의 분지형 스키마는 깊이를 의미하는 n 값을 3으로 고정하고, 각 노드에서의 분지의 수를 의미하는 b 값은 15로 고정하였다. 선형 스키마와는 달리 분지 스키마는 각 노드마다 b개의, 즉 15개의 가지들이 나오므로 선형 스키마에 비해 훨씬 많은 노드를 가진다. 그리고 표 2의 분지형 질의들에서 나타난 바와 같이 분지가 일어나는 깊이를 변화시키면서 실험하였으며, 분지의 깊이를 0부터 2까지 증가시키면 정규경로표현의 수가 6개에서 4개로 줄어든다.

그림 10은 순환형/접합형 스키마를 나타낸다. 순환형/접합형 스키마는 깊이인 n 값을 5로 고정하고, s[0] 노드에서의 분지의 수인 b값은 15로 고정한 후, 접합이 일어나는 깊이를 변화시키면서 실험하였다. 질의는 표 2에 나열하였다. 그림 10에서 깊이 4의 노드들에서 나오는 마지막 에지들 e[4][1]부터 e[4][b]까지는 접합의 깊이에 따라 다른 노드로 향하므로 4개의 에지를 그려 표시하였으며, e[4][1] 에지에 대해서는 어느 깊이의 노드로 향하는지를 1부터 4까지의 번호를 매겨 표시하였다.

마지막으로 3절에서 제시한 예제에 대해서도 실험하였다.

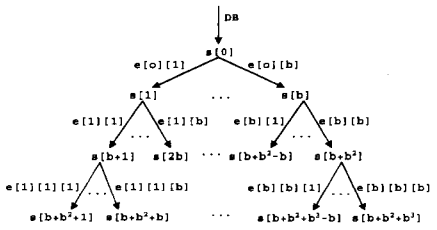


그림 9 분지형 스키마

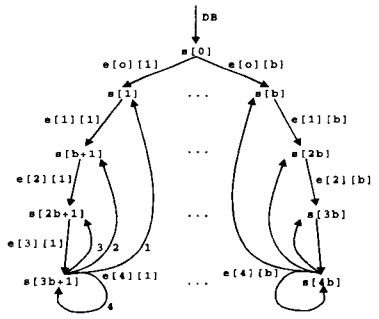


그림 10 순환형/접합형 스키마

**실험결과.** 우선 선형 스키마 및 질의에 대한 실험결과를 그림 11, 그림 12, 그림 13에 보였다. 그림 11은 정규경로표현의 수에 대해서 전체 수행시간이 기존의 경우(그림에서 org로 표시)에는 지수적으로 증가함을 보여준다. 반면에 두 단계 질의전지의 경우(그림에서 2P로 표시)에는 100msec 이하로 안정되어 있음을 보여준다. 이와 같은 경향은 b 값에 무관하나 b 값이 커지면 더욱 두드러지게 나타난다.

그림 12는 전지 단계의 AND/OR 그래프 내의 노드 접근 횟수를 나타낸다. 이는 정규경로표현의 수와 스키마의 크기가 증가함에 따라 노드 접근 횟수가 증가하기는 하지만 지수적으로 증가하지는 않음을 보여준다. 더구나 두 단계 질의전지(2P로 표시)의 전지 단계가 기존의 경우보다 효율적으로 나타남을 유의하기 바란다. 이는 최대 접근 가능성을 더 빨리 구할 수 있기 때문이다. 결국 기존의 경우에 전체 수행시간이 지수적으로 증가하는 것은 후처리 단계에서 확인해야 하는 부분결과들의 조합이 그림 13에서 보인 바와 같이 지수적으로 증가하기 때문이다. 반면에 전처리 단계의 경우 주어진 선형 질의에 대해서 각 정규경로표현을 두 번 정도 확인하면 되며 정규경로표현이 4개라면 총 8회의 확인만으로 최소 최대화 조건 집합을 얻는다. 이러한 경향은 선형 질의를 분지형 스키마나 순환형/접합형 스키마 등의 다른 스키마에 적용해도 동일하게 나타난다.

그림 14는 분지형 스키마에 분지형 질의를 적용한 실험결과이다. 표 2의 분지형 질의들을 살펴보면 알 수 있듯이 정규경로표현의 수는 분지의 깊이가 깊어짐에 따라 적어진다. 따라서 기존의 질의전지 방법은 전체 수행시간이 분지의 깊이가 0에 다가갈수록 지수적으로 증가한다. 반면에 두 단계 질의전지의 전체 수행시간은 모든 깊이에서 2854msec 이하이다. 순환형/접합형의 경우, 그림 15의 실험결과를 보면 접합의 깊이가 얕을수록 두 단계 질의전지의 성능에 근접한다. 이는 접합변수가 둘 이상의 정규경로표현에 의해 공유되어 AND/OR 그래프

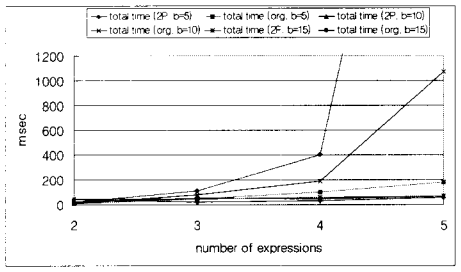


그림 11 전체 수행시간

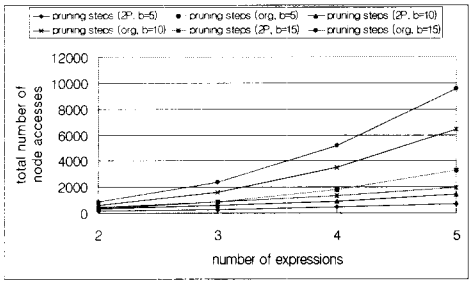


그림 12 전지 단계

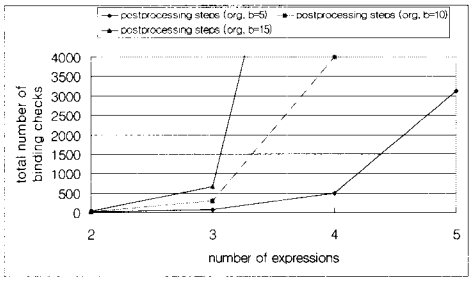


그림 13 후처리 단계

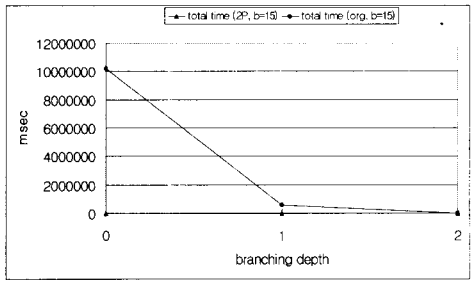


그림 14 분지형의 전체 수행시간

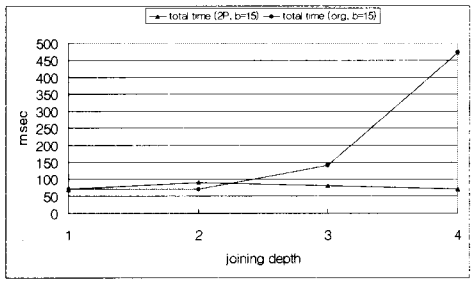


그림 15 순환형/접합형의 전체 수행시간

에서 접근가능한 AND 노드들로 나타나는 부분결과의 수가 급격히 줄기 때문이다.

논문 전체에서 사용한 예제도 실험을 통해 원하는 결과가 나오는지 확인하였다. 접합의 깊이가 1인 접합변수 B에 의해 부분결과의 조합수가 매우 적고 스키마의 크기도 14(노드의 수)로 매우 작기 때문에 두 단계 질의전지는 70msec가 걸리고, 기존의 질의전지는 40msec가 걸린다. 두 단계 질의전지가 상대적으로 더 많은 시간이 걸린 이유는 최대 접근 가능성을 구하는 반복회수가 3으로 동일한 반면에 두 단계 질의전지에서의 노드수가 588개로 기존 방법의 350개보다 다소 많기 때문이다. 예제의 실험결과는 표 3에 요약하였다.

표 3 예제 실험결과 요약

| 정규경로표현의 수 = 5 | | |
|---|---|---|
| 두 단계 질의전지 | 전처리 단계에서의 정규경로표현 접근 회수 | 13 |
| | 전처리 시간 | 0msec |
| | 총 노드 수 | 588 |
| | 반복 수 | 3 |
| | 전지 단계의 노드 접근 회수 | 1764 |
| | 전지 단계 수행시간 | 70msec |
| | 전체 수행시간 | 70msec |
| 기존 질의전지 | 총 노드 수 | 350 |
| | 반복 수 | 3 |
| | 전지 단계의 노드 접근 회수 | 1050 |
| | 전지 단계 수행시간 | 30msec |
| | 후처리 단계에서의 부분결과 확인 회수 | 40 |
| | 후처리 시간 | 10msec |
| | 전체 수행시간 | 40msec |

마지막으로 두 단계 질의전지의 유효성을 보이기 위해 후처리 없이 기존의 질의전지로 구한 질의와 두 단계 질의전지로 구한 질의를 b 값이 15인 실험의 결과로부터 표 4에 나열하였다. 다른 b 값의 경우에도 동등한 결과를 얻기 때문에 불필요한 중복을 피하기 위해 b 값이 15인 실험의 결과만을 포함한다. 그리고 예제 질의의 경우도 앞 절들을 통해 충분히 설명이 되었으므로 불필요한 중복을 피하기 위해 표 4에는 포함시키지 않았다. 기존의 질의전지로도 후처리를 하면 두 단계 질의전지로 구한 것과 동등한 질의를 구할 수 있다. 하지만 두 단계 질의전지의 결과는 전처리를 거치면서 불필요한 매개변수들을 생략하는 경우가 있으므로 기존 질의전지의 결과보다 정규경로표현의 수가 적을 수 있음을 유의하기 바란다.

## 8. 결 론

우리는 본 논문을 통해 PTIME에 스키마 수준 초과경로들을 제거하는 효과적이면서도 확장성이 있는 두 단계 질의전지를 제시하였다. 다중 정규경로표현이 주어질 때 발생하는 기존 질의전지의 문제를 해결하기 위한 새로운 개념에 근거하여, 전처리 단계는 어떤 입력 질의로부터 정규경로표현의 수에 대해서 PTIME에 모든 최소 최대화 조건을 찾는다. 그리고 전지 단계는 기존 질의전지의 확장판으로 정규경로표현들의 상태 수와 준구조적 스키마의 스키마 노드 수에 대해서 PTIME에 완전히 최적화된 경로표현들을 생성한다.

본 논문은 이러한 두 단계 질의전지의 정확성과 유효성, 그리고 확장성을 이론적으로 설명함은 물론 원형 시스템을 이용한 실험결과로도 보였다. 결론적으로, 다중 정규경로표현을 위한 두 단계 질의전지는 이전의 방법들과는 달리 효과적인 동시에 확장성도 있다.

## 참 고 문 헌


[1] Serge Abiteboul, *"Querying Semi-Structured Data,"* Proceedings of the 6th International Conference on Database Theory, pages 1-18, 1997.

[2] Serge Abiteboul, Peter Buneman, and Dan Suciu, *Data on the Web: From Relations to Semistructured Data and XML*, Morgan Kaufmann Publishers, San Francisco, California, USA, 2000.

[3] Peter Buneman, *"Semistructured Data,"* Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pages 51-61, 1997.

[4] Peter Buneman, Susan B. Davidson, Mary F. Fernandez, and Dan Suciu, *"Adding Structure to Unstructured Data,"* Proceedings of the 6th International Conference on Database Theory, pages 336-350, 1997.

[5] Roy Goldman and Jennifer Widom, *"DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases,"* Proceedings of the 23rd International Conference on Very Large Data Bases, pages 436-445, 1997.

[6] Svetlozar Nestorov, Jeffrey D. Ullman, Janet L. Wiener, Sudarshan S. Chawathe, *"Representative Objects: Concise Representations of Semistructured, Hierarchical Data,"* Proceedings of the 13th International Conference on Data Engineering, pages 79-90, 1997.

표 4 실험에 사용한 질의의 최적화된 결과 (b=15)

| 종류 | 질의명 | 후처리를 하지 않은 기존 질의전지 | 두 단계 질의전지 |
|---|---|---|---|
| 선형 | Ql2 | (e[0][1]\|e[0][2]\|...\|e[0][15]).X1 in DB,<br>c[1][1].X2 in X1 | e[0][1].e[1][1].X2 in DB |
| | Ql3 | (e[0][1]\|e[0][2]\|...\|e[0][15]).X1 in DB,<br>(c[1][1]\|c[1][2]\|...\|c[1][15]).X2 in X1,<br>c[2][1].X3 in X2 | e[0][1].e[1][1].e[2][1].X3 in DB |
| | Ql4 | (e[0][1]\|e[0][2]\|...\|e[0][15]).X1 in DB,<br>(c[1][1]\|c[1][2]\|...\|c[1][15]).X2 in X1,<br>(c[2][1]\|c[2][2]\|...\|c[2][15]).X3 in X2,<br>c[3][1].X4 in X3 | e[0][1].e[1][1].e[2][1].e[3][1].X4 in DB |
| | Ql5 | (e[0][1]\|e[0][2]\|...\|e[0][15]).X1 in DB,<br>(c[1][1]\|c[1][2]\|...\|c[1][15]).X2 in X1,<br>(c[2][1]\|c[2][2]\|...\|c[2][15]).X3 in X2,<br>(e[3][1]\|e[3][2]\|...\|e[3][15]).X4 in X3,<br>c[4][1].X5 in X4 | e[0][1].e[1][1].e[2][1].e[3][1].e[4][1].X5 in DB |
| 분지형 | Qb0 | (e[1]\|e[2]\|...\|e[15]).X1 in DB,<br>(c[1][1]\|c[1][2]\|...\|c[15][15]).X12 in X1,<br>c[1][1][1].X13 in X12,<br>(c[1]\|c[2]\|...\|c[15]).X2 in DB,<br>(c[1][1]\|c[1][2]\|...\|c[15][15]).X22 in X2,<br>c[15][15][15].X23 in X22 | e[1].e[1][1].e[1][1][1].X13 in DB,<br>c[15].c[15][15].c[15][15][15].X23 in DB |
| | Qb1 | (e[1]\|e[2]\|...\|e[15]).X1 in DB,<br>(c[1][1]\|c[1][2]\|...\|c[15][15]).X12 in X1,<br>c[1][1][1].X13 in X12,<br>(c[1][1]\|c[1][2]\|...\|c[15][15]).X22 in X1,<br>c[1][15][15].X23 in X22 | e[1].X1 in DB,<br>c[1][1].c[1][1][1].X13 in X1,<br>c[1][15].c[1][15][15].X23 in X1 |
| | Qb2 | (e[1]\|e[2]\|...\|e[15]).X1 in DB,<br>(c[1][1]\|c[1][2]\|...\|c[15][15]).X12 in X1,<br>c[1][1][1].X13 in X12,<br>c[1][1][15].X23 in X12 | e[1].e[1][1].X12 in DB,<br>c[1][1][1].X13 in X12,<br>c[1][1][15].X23 in X12 |
| 순환형 또는 접합형 | Qj1 | (e[0][1]\|e[0][2]\|...\|e[0][15]).X1 in DB,<br>c[1][1].X2 in X1,<br>c[2][1].X3 in X2,<br>c[3][1].X4 in X3,<br>c[4][1].X1 in X4 | e[0][1].X1 in DB,<br>c[1][1].c[2][1].c[3][1].c[4][1].X1 in X1 |
| | Qj2 | (e[0][1]\|e[0][2]\|...\|e[0][15]).X1 in DB,<br>(c[1][1]\|c[1][2]\|...\|c[1][15]).X2 in X1,<br>c[2][1].X3 in X2,<br>c[3][1].X4 in X3,<br>c[4][1].X2 in X4 | e[0][1].e[1][1].X2 in DB,<br>c[2][1].c[3][1].c[4][1].X2 in X2 |
| | Qj3 | (e[0][1]\|e[0][2]\|...\|e[0][15]).X1 in DB,<br>(c[1][1]\|c[1][2]\|...\|c[1][15]).X2 in X1,<br>(c[2][1]\|c[2][2]\|...\|c[2][15]).X3 in X2,<br>c[3][1].X4 in X3,<br>c[4][1].X3 in X4 | e[0][1].e[1][1].e[2][1].X3 in DB,<br>c[3][1].c[4][1].X3 in X3 |
| | Qj4 | (e[0][1]\|e[0][2]\|...\|e[0][15]).X1 in DB,<br>(c[1][1]\|c[1][2]\|...\|c[1][15]).X2 in X1,<br>(c[2][1]\|c[2][2]\|...\|c[2][15]).X3 in X2,<br>(c[3][1]\|c[3][2]\|...\|e[3][15]).X4 in X3,<br>c[4][1].X4 in X4 | e[0][1].e[1][1].e[2][1].e[3][1].X4 in DB,<br>c[4][1].X4 in X4 |

[7] Mary F. Fernandez and Dan Suciu, *"Optimizing Regular Path Expressions Using Graph Schemas,"* Proceedings of the 14th International Conference on Data Engineering, pages 14-23, 1998 (The full version is available at http://www. cs.washington. edu/homes/suciu/files/paper-techrep.ps).

[8] Jason McHugh and Jennifer Widom, *"Compile-Time Path Expansion in Lore,"* Proceedings of the Workshop on Query Processing for Semi-structured Data and Non-Standard Data Formats, 1999.

[9] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Moshe Y. Vardi, *"Rewriting of Regular Expressions and Regular Path Queries,"* Proceedings of the 18th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pages 194-204, 1999.

[10] Jason McHugh and Jennifer Widom, *"Optimizing Branching Path Expressions,"* Technical Report, Stanford University, 1999.

[11] Michael Kifer, Won Kim, and Yehoshua Sagiv, *"Querying Object-Oriented Databases,"* Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 393-402, 1992.

[12] Vassilis Christophides, Serge Abiteboul, Sophie Cluet, and Michel Scholl, *"From Structured Documents to Novel Query Facilities,"* Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 313-324, 1994.

[13] Dallan Quass, Anand Rajaraman, Yehoshua Sagiv, Jeffrey D. Ullman, and Jennifer Widom, *"Querying Semistructured Heterogeneous Information,"* Proceedings of the 4th International Conference on Deductive and Object-Oriented Databases, pages 319-344, 1995.

[14] Serge Abiteboul, Dallan Quass, Jason McHugh, Jennifer Widom, and Janet L. Wiener, *"The Lorel Query Language for Semistructured Data,"* International Journal on Digital Libraries, 1(1), pages 68-88, 1997.

[15] Alon Y. Halevy, *"Theory of Answering Queries Using Views,"* SIGMOD Record, 29(4), 2000.

[16] Jason McHugh and Jennifer Widom, *"Query Optimization for XML,"* Proceedings of the 25th International Conference on Very Large Data Bases, pages 315-326, 1999.

[17] Vassilis Christophides, Sophie Cluet, and Guido Moerkotte, *"Evaluating Queries with Generalized Path Expressions,"* Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 413-422, 1996.

박 창 원

1992년 인하대학교 전자공학과(학사). 1992년 ~ 1993년 LG전자 VIDEO 연구소 연구원. 1995년 서강대학교 전자계산학과(학사). 1997년 한국과학기술원 전산학과(석사). 1997년 ~ 현재 한국과학기술원 전산학과 박사 과정. 2002년 ~ 현재 LG전자기술원 정보기술연구소 선임연구원. 관심분야는 XML 질의처리 및 최적화, XML 데이타 관리, Web 데이타 관리, GIS, LBS

정 진 완

1973년 서울대학교 공과대학 전기공학과(학사). 1983년 University of Michigan 컴퓨터 공학과(박사). 1983년 ~ 1993년 미국 GM 연구소 전산과학과 선임연구원 및 책임연구원. 1993년 ~ 현재 한국과학기술원 전산학과 부교수 및 교수. 관심분야는 XML, 멀티미디어 데이타베이스, GIS, 웹 정보검색, 객체지향