소수 레이블을 이용한 RDF/RDFS 인덱스 구조*

김선영°, 권동섭, 이석호
서울대학교 전기컴퓨터공학부
{harpist1°, subby}@db.snu.ac.kr, shlee@cse.snu.ac.kr

Indexing Scheme for RDF/RDFS using Prime Number Label

Sunyoung Kim°, Dongseop Kwon, Sukho Lee
School of Electrical Engineering and Computer Science, Seoul National University

요 약

시맨틱 웹의 등장에 따라 RDF와 RDF Schema(RDF/RDFS)로 표현되는 웹 데이타의 양이 증가하고 있다. 이에 웹 데이타를 효율적으로 저장, 검색할 수 있는 인덱스 구조의 필요성이 높아지고 있다. 본 연구에서는 기존의 트리 모델을 위한 소수 레이블 기법(prime number labeling scheme)을 발전시켜, RDF/RDFS 인덱스 구조를 표현할 수 있는 그래프 모델을 위한 소수 레이블 기법을 제안한다. 제안한 기법은 기존의 소수 레이블 기법을 그래프에 적용하여 구조 질의(structural query)를 효율적으로 처리할 수 있고, 데이타 갱신 시에 인덱스를 재구성하지 않아도 되는 장점을 가지고 있다. 그리고 이전의 RDF/RDFS 인덱스 구조에서 효율적으로 처리하기 힘들었던 순환 방향성 그래프에 대한 질의도 쉽게 처리할 수 있다.

1. 서론

시맨틱 웹(Semantic Web)은 웹 데이타에 잘 정의된 의미(Semantic)를 부여하여, 사람뿐만 아니라 컴퓨터도 그 의미를 해석할 수 있는 차세대 웹이다. 시맨틱 웹을 통해 원하는 정보를 정확하고 효율적으로 검색하고, 그 정보들을 통합, 재사용할 수 있다. Resource Description Framework(RDF)[1]은 데이타의 메타 데이타를 기술하는 모델로, 시맨틱 웹 데이타의 의미를 정의하는 기본적인 언어이다. 그리고 RDF Schema(RDFS)는 RDF로 표현할 수 없는 데이타간의 관계를 클래스와 속성으로 표현한다.

RDF와 RDF Schema(RDF/RDFS)로 표현되는 웹 데이타의 양이 빠른 속도로 증가함에 따라, RDF/RDFS를 효율적으로 저장하고, 검색할 수 있는 인덱스 구조의 필요성이 높아지고 있다. 특히 데이타의 값을 얻는 단순한 질의뿐만 아니라 데이타간의 관계를 판별하는 구조 질의(structural query)를 효과적으로 처리할 수 있는 인덱스 구조가 필요하다. 그러나 현재 RDF/RDFS의 인덱스 구조로는 구조 질의를 쉽게 처리할 수 없다.

본 논문에서는 XML에서 사용되는 소수 레이블 기법(prime number labeling scheme)[2]을 이용한 RDF/RDFS 인덱스 구조를 제안한다. 본 논문에서 사용하는 소수 레이블 기법은 XML 트리를 위한 소수 레이블 기법을 그래프에 맞게 확장한 것이다. 제안한 기법을 사용하면 RDF/RDFS에서도 구조 질의를 쉽게 처리할 수 있고, 데이타 갱신 시에 인덱스를 재구성하지 않아도 된다. 그리고 순환을 포함하는 그래프에 대한 질의도 효율적으로 처리할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로 RDF/RDFS 인덱스 구조에 대해 알아본다. 3장에서는 XML 트리 모델을 위한 소수 레이블 기법을 소개하고, RDF/RDFS 그래프에 맞게 확장한 소수 레이블 기법을 제안한다. 4장에서는 제안한 기법에 따른 RDF/RDFS 인덱스 구조를 제시하고, 5장에서 결론을 맺는다.

2. 관련 연구

RDF/RDFS를 위한 가장 간단한 인덱스 방법은 기존의 XML 인덱스 구조를 사용하는 것이다. 기본적으로 RDF의 문법은 XML 문법을 따르므로, XML 인덱스 구조를 사용하면 구현이 간단하다. 그러나 RDF는 XML과는 다른 특성을 가진다. 즉, RDF 데이타는 트리가 아닌 방향성 그래프로 모델링되며, 노드뿐만 아니라 간선도 값을 가진다. 그러므로 RDF 데이타를 XML 인덱스 구조로 구현하는 방법은 적합하지 않다.

현재 많이 사용되는 방법으로는 크게 관계형 데이타베이스 시스템을 이용하는 방법과 접미사 배열 인덱스 구조를 이용하는 방법이 있다. 관계형 데이타베이스를 이용하는 시스템 중 하나인 RDFSuite[3]는 RDF/RDFS를 객체 관계형 구조에 맞게 분할하여, 해당 테이블에 저장한다. 그러나 RDFSuite는 데이타를 객체 중심으로 저장하므로, 조상-자손 관계 질의와 같은 구조 질의 처리에 어려움이 있다. 그리고 유일하지 않은 값인 속성을 중심으로 저장하기 때문에, 검색할 때에 질의가 원하는 속성에 해당하는 복수개의 튜플을 검색해야 한다.

접미사 배열 인덱스 구조[4]를 이용하는 방법은 비순환 방향성 그래프의 경로 표현을 중심으로 RDF/RDFS의 인덱스를 구축하는 것으로, 모든 경로의 접미사 배열을 검색에 사용하기 때문에 경로 질의 처리에 효율적이다. 그러나 이 방법 역시 조상-자손 관계와 같은 구조 질의를 처리하기 어렵고, RDF/RDFS 데이타가 갱신될 때마다 경로와 그에 따른 접미사 배열을 처음부터 다시 생성해야 한다. 그리고 RDF/RDFS 데이타가 비순환 방향성 그래프인 경우로 한정되기 때문에, 순환 방향성 그래프로 모델링 되는 경우에는 질의를 나눠 처리해야 하는 문제가 발생할 수 있다.

3. 그래프 모델을 위한 소수 레이블 기법

이 장에서는 그래프 모델을 위한 소수 레이블 기법에 대해 설명한다. 3.1절에서는 기존의 트리 모델을 위한 소수 레이블 기법을 소개하고, 이를 바탕으로 RDF/RDFS 그래프를 위한 소수 레이블 기법을 제안한다. 그래프 모델을 위한 소수 레이블 기법은 3.2절의 비순환 방향성 그래프인 경우와 3.3절의 순환 그래프인 경우로 나누어 단계적으로 확장하여 제시한다.

3.1 트리 모델을 위한 소수 레이블 기법

XML의 소수 레이블 기법[2]은 구조 질의 처리에 효율적이고, 데이타 갱신에 따른 인덱스 재구성에 대한 제약이 적다. 트리 모델을 위한 소수 레이블 기법은 다음과 같다.

* 본 연구는 2005년도 두뇌한국21사업과 정보통신부의 대학 IT연구센터(ITRC) 지원을 받아 수행되었습니다.

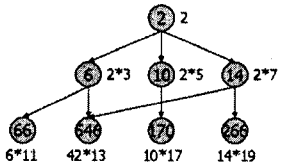
① 각 노드에 소수를 하나씩 할당한다. 이 때 할당된 소수를 셀프 레이블(Self Label)이라 한다.
② 각 노드의 소수 레이블(Prime Number Label)을 계산한다. 소수 레이블은 해당 노드의 부모 노드의 소수 레이블에 셀프 레이블을 곱한 값이다.

3.2 비순환 방향성 그래프를 위한 소수 레이블 기법

RDF/RDFS는 트리가 아닌 방향성 그래프로 모델링 되므로, 위의 트리를 위한 소수 레이블 기법을 확장한 기법을 사용한다. 먼저, 순환이 포함되지 않은 방향성 그래프를 위한 소수 레이블 기법에 대해 살펴보면 다음과 같다. RDF/RDFS 그래프는 하나의 연결된 컴포넌트로 구성되어 있다고 가정한다.

- 부모 노드가 하나인 경우에는 3.1절의 트리 모델을 위한 소수 레이블 기법을 사용한다.
- 부모 노드가 둘 이상인 경우에는 부모 노드들의 소수 레이블의 최소 공배수에 셀프 레이블을 곱한 값이 소수 레이블이 된다. 즉, 소수 레이블은 부모 노드들의 소수 레이블을 모두 반영해야 하는 조건을 만족시키는 값 중에서 가장 작은 값이다.

그림 1은 비순환 방향성 그래프에 소수 레이블 기법을 적용한 예를 그래프로 나타낸 것이다.



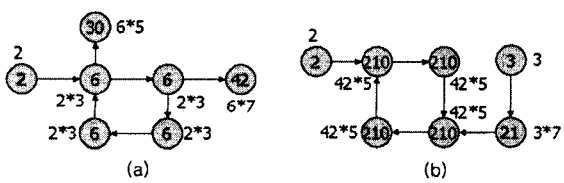
[그림 1] 비순환 방향성 그래프을 위한 소수 레이블 기법

3.3 순환 방향성 그래프를 위한 소수 레이블 기법

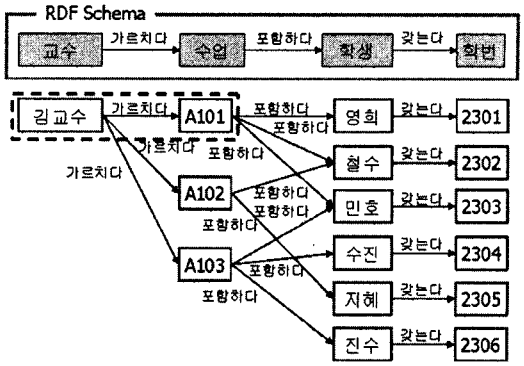
RDF/RDFS를 모델링한 그래프에 순환이 존재하는 경우에는 [5]와 유사한 방법을 사용하여, 순환에 포함되는 노드들에 특별한 소수 레이블 기법을 적용한다. 순환 방향성 그래프를 위한 소수 레이블 기법을 살펴보면 다음과 같다.

- 순환에 포함되지 않는 노드는 3.2절의 비순환 방향성 그래프를 위한 소수 레이블 기법을 사용한다.
- 하나의 순환에 포함되는 모든 노드는 같은 셀프 레이블을 할당한다. 이 노드들이 집합 A에 속한다고 하면,
  - 집합 A에 대한 부모 노드가 하나인 경우에는 3.2절의 첫 번째 방법으로 소수 레이블을 구한다.
  - 집합 A에 대한 부모 노드가 둘 이상인 경우에는 3.2절의 두 번째 방법으로 소수 레이블을 구한다.

그림 2는 순환 방향성 그래프에 소수 레이블 기법을 적용한 예이다. 그림 2의 (a)는 하나의 순환에 속하는 전체 노드에 부모 노드가 하나인 경우이고, (b)는 부모 노드가 둘인 경우이다.



[그림 2] 순환 방향성 그래프를 위한 소수 레이블 기법



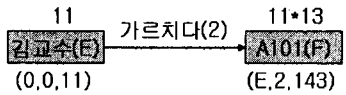
[그림 3] RDF/RDFS 그래프의 예

4. 소수 레이블을 이용한 RDF/RDFS 인덱스 구조

이 장에서는 3장의 소수 레이블을 이용한 RDF/RDFS의 인덱스 구조를 제안한다. 4.1절에서는 소수 레이블을 이용한 RDF/RDFS의 인덱스 구조와 구축을 설명하고, 4.2절에서는 소수 레이블을 이용한 RDF/RDFS 저장 구조를 제시한다. 그리고 4.3절에서는 제안하는 인덱스 구조에서의 질의 처리에 대해 설명한다.

4.1 소수 레이블을 이용한 RDF/RDFS의 인덱스 구조와 구축

RDF/RDFS의 한 문장은 노드인 주어와 목적어, 그리고 주어와 목적어를 잇는 간선인 술어로 이루어져 있다. 그림 4는 그림 3의 RDF/RDFS 그래프 중에서 점선 부분인 "김교수는 A101을 가르친다." 는 문장을 RDF 그래프로 나타내고, 각 노드의 인덱스 구조를 표시한 것이다.



[그림 4] RDF 문장의 인덱스 구조

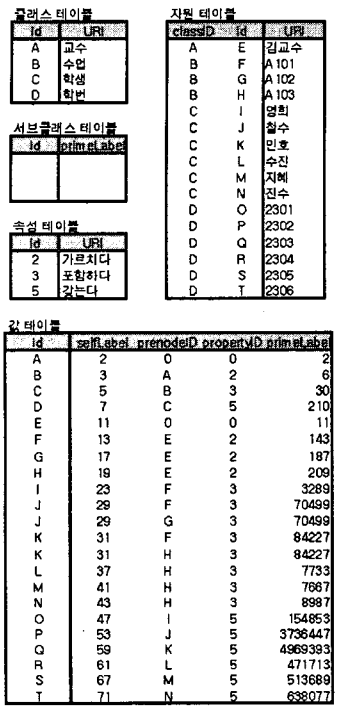
하나의 노드는 이전 노드의 ID와 속성 ID, 그리고 자신의 소수 레이블을 값으로 갖는다(prenodeID, propertyID, primeLabel). 즉, 노드는 자신이 목적어가 되는 한 문장의 값을 인덱스로 가진다. 그리고 소수 레이블은 노드 간의 관계를 나타내는 역할을 한다.

소수 레이블을 이용한 RDF/RDFS의 인덱스 구조를 구축하는 방법은 다음과 같다.

① RDF/RDFS 문서를 방향성 그래프로 변환한다.
② 넓이 우선 탐색(BFS)를 하면서,
  - 노드의 id와 셀프 레이블을 할당한다.
  - 노드의 소수 레이블을 계산한다.

4.2 소수 레이블을 이용한 RDF/RDFS의 저장 구조

소수 레이블을 이용한 RDF/RDFS의 저장 구조는 클래스 테이블, 서브클래스 테이블, 속성테이블, 자원 테이블, 그리고 값 테이블로 구성된다. 그림 5는 그림 3의 RDF/RDFS 그래프를 RDF/RDFS 인덱스 구조로 변환하여 데이타베이스에 저장한 것이다.



[그림 5] RDF/RDFS의 저장 구조

클래스 테이블과 속성 테이블은 각각 RDFS에 정의된 클래스와 속성에 대한 URI를 저장한다. 자원 테이블은 RDF에 있는 자원의 URI와 함께 그 자원이 RDFS의 어느 클래스에 속하는지에 대한 정보를 저장한다. 값 테이블은 4.1절에서 설명한 바와 같이 소수 레이블을 이용한 RDF/RDFS 인덱스 구조를 저장한 것으로, RDF/RDFS에 존재하는 모든 클래스와 자원에 대한 정보를 가지고 있다. 값 테이블에서 튜플 하나는 하나의 노드를 나타냄과 동시에 해당 노드를 목적어로 갖는 하나의 RDF 문장을 저장한다.

마지막으로 서브클래스 테이블은 클래스 또는 자원간의 슈퍼클래스와 서브클래스 관계를 저장하는 테이블로, 값 테이블에서의 소수 레이블과는 별도의 소수 레이블을 갖는다. 이는 클래스 또는 자원간의 관계를 나타내는 속성을 두 종류로 나누어 저장했기 때문이다. 즉, 슈퍼클래스, 서브클래스와 같은 상속 속성에 관한 질의를 처리하려면 서브클래스 테이블을 이용하고, 상속 이외의 속성에 대한 질의를 처리하려면 값 테이블을 이용하면 된다.

RDFSuite와 비교했을 때 저장 구조면에서 다른 점은 서브속성 테이블이 존재하지 않는다는 점이다. 본 논문에서 제안하는 RDF/RDFS 저장 구조의 속성 테이블은 테이블 자체에 소수 레이블링을 도입하였다. 클래스 혹은 자원간의 관계는 상속과 그 밖의 여러 가지 속성들로 표현되지만, 속성간의 관계는 상속만 존재한다. 그러므로 별도로 서브속성 테이블을 두지 않고, 속성 테이블만으로 속성간의 상속 관계를 표현할 수 있다.

4.3 소수 레이블을 이용한 RDF/RDFS 인덱스 구조의 질의 처리

이 절에서는 소수 레이블을 이용한 RDF/RDFS 인덱스 구조에 대한 질의 처리 과정을 예를 통해 설명한다.

[질의 1] 부모-자식 관계

"수업 A102는 민호를 포함하는가?"

질의 1은 부모-자식 관계 질의로, 두 가지 방법을 통해 질의 처리를 할 수 있다. 한 가지 방법은 목적어에 해당하는 노드의 인덱스에서 이전 노드 값이 질의에 주어진 주어의 값과 동일한지 비교하는 방법이다. 예에서 민호의 이전 노드 값은 F와 H이고 A102는 G이므로, 질의의 답은 거짓이 된다. 다른 방법은 두 노드를 찾아 각 노드의 소수 레이블을 비교하는 방법이다. 예에서 민호의 소수 레이블 84227은 A102의 소수 레이블 6으로 나누면 나머지가 생기므로, 질의의 답은 거짓이 된다. 하나의 튜플은 하나의 노드를 나타냄과 동시에 하나의 문장을 저장하고 있으므로, 부모-자식 관계 질의를 처리할 때에는 굳이 소수 레이블을 비교하는 후자의 방법을 사용할 필요는 없다.

[질의 2] 조상-자손 관계

"김교수가 가르치는 수업 중에서 학번이 2303인 학생이 듣는 수업은 무엇인가?"

질의 2는 조상-자손 관계 질의이다. 제일 먼저 김교수와 학번이 2303인 학생이 조상-자손 관계가 성립하는지 확인해야 한다. 이전의 RDF/RDFS 인덱스 구조를 사용하면 두 노드의 조상-자손 관계 여부를 확인하기 위해 여러 번의 조인 연산을 하거나 불필요한 검색을 수행해야 한다. 하지만 제안하는 기법에서는 두 노드의 소수 레이블만을 확인하면 된다. 예에서 학번 2303의 소수 레이블 4969393은 김교수의 소수 레이블 11로 나누면 나머지가 0이므로, 조상-자손 관계가 성립한다는 사실을 알 수 있다. 그리고 김교수를 이전 노드로 갖는 노드와 학번 2303의 소수 레이블을 확인하여 조상-자손 관계를 갖는 노드가 답이므로, 질의의 답은 A101과 A103이다.

5. 결론

본 논문에서는 기존의 트리 모델을 위한 소수 레이블 기법을 발전시켜, RDF/RDFS 인덱스 구조를 표현할 수 있는 소수 레이블 기법을 제안하였다. 제안한 기법은 기존의 소수 레이블 기법을 그래프 모델에 적용하였기 때문에 RDF/RDFS 데이타에 대해 구조 질의를 효율적으로 처리할 수 있고, 데이타 갱신 시에 인덱스를 재구성하지 않아도 된다. 그리고 이전의 RDF/RDFS 인덱스 구조에서 효율적으로 처리하기 어려운 순환 방향성 그래프에 대한 질의도 쉽게 처리할 수 있다.

참고문헌

[1] Pierre-Antoine Champin, "RDF Tutorial", http://www710.univ-lyon1.fr/~champin/rdf-tutorial/rdf-tutorial.html.

[2] Xiaodong Wu, Mong Li Lee, Wynne Hsu, "A Prime Number Labeling Scheme for Dynamic Ordered XML Trees", Proceedings of the 20th International Conference on Data Engineering(ICDE'04), 2004.

[3] Sofia Alexaki, Vassilis Christophides, Greg Karvounarakis, Dimitris Plexousakis, Karsten Tolle, "The RDFSuite: Managing Voluminous RDF Description Bases", Semantic Web Workshop 2001, 2001.

[4] Akiyoshi Matono, Toshiyuki Amagasa, Masatoshi Yoshikawa, Shunsuke Uemura, "An Indexing Scheme for RDF and RDF Schema based on Suffix Arrays", Semantic Web DataBase Workshop 2003, 2003.

[5] Hongzhi Wang, Wei Wang, Xuemin Lin, Jianzhong Li, "Labeling Scheme and Structural Joins for Graph-Structured XML Data", 7th Asia-Pacific Web conference(APWeb2005), Shanghai, China, 2005.