동적 XML 환경에서의 원형 레이블링 방법

김진영○ 박 석
서강대학교 컴퓨터공학과
{bubble, spark}@sogang.ac.kr

Circle Labeling Scheme for Dynamic XML Data

Jinyoung Kim○ Seog Park
Department of Computer Science and Engineering, Sogang University

1. 서 론

스트림 데이터를 사용하는 유비쿼터스 환경과 인터넷 환경에서 XML은 데이터의 저장과 교환, 출판에 있어 광범위하게 표준으로 사용되고, XML의 효율적이고 안전한 사용을 위해 레이블링 방법[1-3] 에 대한 다양한 연구들이 제시되고 있다.

최근 연구[1-3]에서는 전체 레이블에 대한 큰 저장공간이 필요한 문제점이 있고, 동적 XML 환경에서 같은 위치에 지속적으로 데이터가 삽입될 경우 레이블 길이가 증가하거나 레이블 갱신에 대한 추가적인 계산 비용이 발생하는 문제점이 있다. 레이블링 방법은 다음과 같은 요구 조건을 갖는다.

(1) 레이블로 XML 문서 내 엘리먼트들의 순서 표현
(2) 레이블로 조상-자손, 형제 노드 관계 표현
(3) XML 문서 내 전체 레이블에 대한 작은 저장공간
(4) 동적 XML 환경에 대한 고려 사항으로 기존 XML 문서에 대한 갱신 연산 발생 시 전체 레이블에 대한 갱신 없이 해당 데이터의 레이블만 갱신

제안하는 원형 레이블링 방법(CLS:Circle Labeling Scheme)은 이차원 인덱스에 기반하며 XML 문서 내 데이터 간 순서와 조상-자손, 형제 노드 관계를 표현할 수 있어 탐색 시간을 줄일 수 있고, XML 트리의 깊이에 관계 없이 일정한 레이블 길이를 유지할 수 있어 저장공간을 효율적으로 사용할 수 있다.

XML 문서를 원으로 표현하고 회전수 개념을 적용하여 레이블 구조 내에 필요한 변수의 크기를 줄이고, 루트원 / 확장원 개념을 적용함으로써 레이블 값의 누적을 방지하여 작은 크기의 변수를 사용할 수 있게 함으로써 큰 XML 문서 내의 레이블 저장공간의 효율을 얻을 수 있다. 또한, 동적 XML 환경에 대한 고려로 반지름 개념을 적용하여 기존 연구들에서 레이블이 길어지는 문제점을 해결하였고 레이블 저장공간의 효율을 얻을 수 있다.

2. 제안하는 방법 : 원형 레이블링 방법(CLS)

2.1 이차원 인덱스 : 기본 레이블

CLS는 이차원 인덱스에 기반한다. XML 문서의 첫 태그를 시작으로 1부터 증가하는 수를 부여해 각 엘리먼트의 시작, 끝 태그의 값을 쌍으로 묶어 (start, end) 값을 생성한다.

(start, end) 레이블을 사용하여 다음의 두 조건을 바탕으로 조상-자손, 형제 노드 관계를 파악할 수 있다.

(1) 어떤 노드 u, v에 대해, u와 v의 레이블이 ($start_u$,end_u), ($start_v$,end_v)일 때, "$start_u$<$start_v$∧end_u>end_v" 이면 u는 v의 조상노드

(2) 어떤 노드 u, v에 대해, u와 v의 레이블이 (startu,endu), ($start_v$,end_v)일 때, "$start_u$<end_u<$start_v$<end_v" 이면 u는 v의 전임자노드

2.2 회전수(RN)을 사용한 원형 레이블링 방법

[그림 1] 에서는 이차원 인덱스 사용 시 XML 문서 크기 증가에 따라 레이블에 사용되는 기존 변수 크기를 증가시켰을 때, 불필요하게 큰 변수를 사용함으로써 내부단편화가 발생하여 저장공간의 낭비가 발생함을 나타낸다.

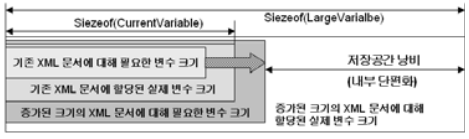


그림 1. 이차원 인덱스 사용 시 저장공간 낭비의 예

2.2.1 회전수(RN)을 사용한 원형 레이블링 방법

XML 문서는 [그림 2(a)]에서처럼 이차원 인덱스로 레이블링 할 수 있다. [그림 2(b)]는 [그림 6(a)]의 레이블을 수평선에 표현한 것이다. 본 연구에서는 XML 문서를 수평선에 나타내는 개념이 아닌 [그림 3]에서처럼 원 개념으로 이해하였다.

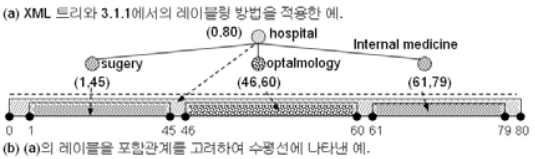


그림 2. XML 문서를 수평선에 표현

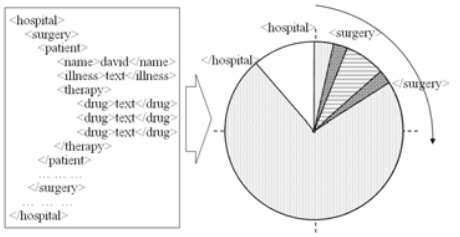


그림 3. XML 문서를 원으로 표현

원 개념을 적용한 CLS의 기본 레이블 구조는 다음과 같다.
(start_angle, end_angle)
- start_angle : 엘리먼트 시작 태그에 대한 각 값
- end_angle : 엘리먼트 끝 내그에 대한 각 값
- 각 값의 범위는 관리자가 지정할 수 있다.
  (실제 원은 360도이지만 논리적으로 확장 축소 가능)

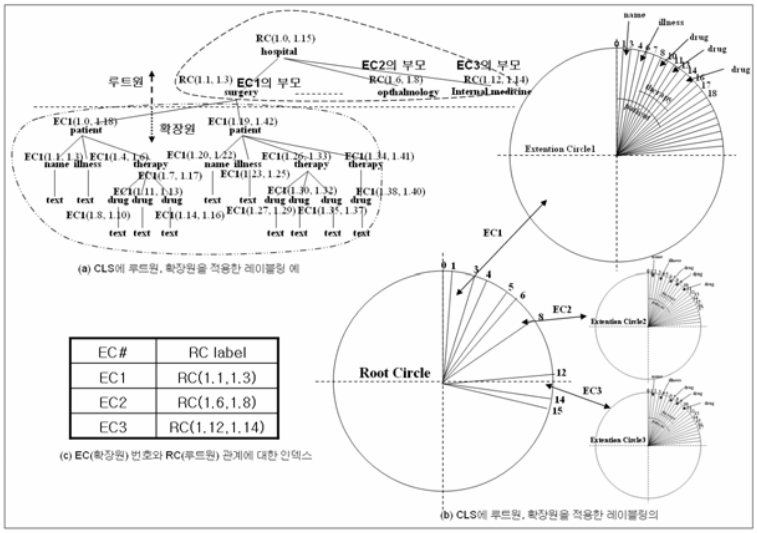
2.2.2 원형 레이블링 방법에 회전 수(RN) 적용

[그림 1]에서처럼 XML 문서의 크기가 증가하면 레이블 구조 내 변수를 더 큰 크기의 변수로 확장해야 하는데 전체 레이블 저장 공간이 불필요하게 커지는 문제점이 발생한다. 정수형 변수를 사용할 경우 65,535의 수를 표현할 수 있고 3만개 정도의 엘리먼트에 대해 레이블링을 할 수 있다. 새로운 데이터의 지속적인 삽입으로 엘리먼트 수가 4만 개로 증가한다면 정수형 변수를 단정도부동소수점 변수로 확장해주어야 한다. 변수 크기의 증가로 전체 레이블 저장 공간은 두 배로 증가한다. 실제로 변수에 대해 몇 비트만 확장해주면 되지만 큰 크기의 변수로 확장함으로써 저장공간의 낭비가 발생한다.

엘리먼트 수가 증가되어 변수 크기를 확장해야 할 때 기존 레이블 변수를 사용하면서 회전수(Rotation Number) 개념을 추가함으로써 저장공간의 효율을 얻을 수 있다.

(start_angle, end_angle)에 회전 수 개념을 적용한 레이블은 **($start\_RN.start\_angle_R$, $end\_RN.end\_angle_R$)** 이다. start_RN (end_RN) 은 관리자가 지정한 최대 각 값으로 start_angle (end_angle)을 나눈 몫이고, $start\_angle_R$(end_angle_R)은 나머지이다. start_angle은 end_angle보다 항상 작기 때문에 start_RN은 end_RN보다 작거나 같다. start_RN(end_RN)의 변수 크기와 $start\_angle_R$(end_angle_R)의 변수 크기의 합은 start_angle (end_angle)의 변수 크기보다 작아야 한다. 이 때, start_RN (end_RN)의 변수 크기가 start_angleR(end_angleR)의 변수 크기보다 항상 작아야 한다.

* 본 연구는 한국과학재단 특정기초연구(R-01-2006-000-10609-0) 지원으로 수행되었음.



EC#	RC label
EC1	RC(1.1,1.3)
EC2	RC(1.6,1.8)
EC3	RC(1.12,1.14)

그림 4. 원형 레이블링 방법에 루트원 / 확장원의 개념을 적용한 예

2.3 루트원(RC) / 확장원(EC) 개념 적용

큰 XML 문서에 대한 레이블링 시 레이블 값이 누적되기 때문에 큰 크기의 변수를 사용해야 한다. XML 문서 내에 자주 나타나는 데이터 묶음(subtree)들에 대해 별도 관리를 함으로써 저장공간의 효율을 얻을 수 있다.

CLS에 루트원(Root Circle)과 확장원(Extention Circle)의 개념을 적용한다. 확장원을 구성하기 전에 자주 사용되는 서브트리의 루트 노드를 결정한다. [그림 4(a)]는 XML 문서에서 루트원의 범위와 확장원의 범위를 정한 뒤 레이블을 할당해준 예이다. [그림 4(b)]에서는 [그림 4(a)]의 레이블을 논리적으로 나타낸 그림이다. [그림 4(c)]는 확장원 식별자로 표현된 확장원의 식별자와 확자원의 루트 노드의 루트원 내의 레이블 간의 관계 테이블이다.

루트원과 확장원 개념을 추가한 원형 레이블링 방법의 레이블 구조는 다음과 같다.

[RC# ¦ EC#].($s\_RN.start\_angle_R$, $e\_RN.end\_angle_R$)

- 루트원과 확장원의 식별자인 RC#, EC#에 대한 변수 크기는 $start_angle_R$과 end_angle_R의 변수보다 작다.

만약 루트원에서 확장원을 검색한다면 루트원의 단말 노드에 대해 루트원 / 확장원 테이블을 검색, 확인할 수 있다. 확장원에서 루트원으로의 검색은 확장원 식별자를 통해 루트원 / 확장원 테이블을 검색하여 해당 확장원의 루트가 되는 노드의 루트원에서의 레이블을 찾음으로써 가능하다.

원형 레이블링 방법에서 루트원 / 확장원 개념을 적용함으로써 큰 XML 문서 내에서 레이블 값의 누적을 방지하여 적은 크기의 변수를 레이블 구조에서 사용함으로써 전체 레이블에 대한 적은 저장공간을 사용할 수 있다.

3. 동적 XML 환경에서의 원형 레이블링 방법(CLS)

XML 문서에 대한 갱신(update) 연산은 삽입(insert), 갱신(modify), 삭제(delete)가 있다. 갱신은 레이블에 상관 없이 데이터의 변경만 고려하면 되고, 삭제도 레이블에 상관 없이 데이터와 레이블을 함께 삭제해 주면 되기 때문에 전체 레이블에 영향을 주지 않는다.

하지만, 새로운 데이터의 삽입 시에는 기존 전체 레이블을 다시 작성해 주어야 하는 문제점이 발생하기 때문에 전체 레이블에 대한 변경 없이 새로 삽입되는 데이터에 대해서만 레이블링할 수 있어야 한다. 기존 연구[1-3]의 경우 같은 위치에 지속적으로 새로운 데이터를 삽입하는 경우 레이블의 길이가 길어지기 때문에 레이블 저장공간의 효율을 떨어뜨리는 문제가 발생한다.

CLS에 반지름 개념을 적용하여 언급했던 문제점들을 해결할 수 있고, 다음과 같은 레이블 구조를 제안한다.

Radius.(start_RN.start_angle, end_RN.end_angle)

- Radius : 초기 값 1, 소수를 표현할 수 있는 변수 사용

시스템이 시작되면 기존 XML문서에 대해 CLS를 통해 전체 레이블을 초기화하는데 이 때 반지름은 1로 초기화한다. 새로운 데이터가 XML 문서에 삽입되면 1) 삽입될 위치를 찾고, 2) 기준이 되는 노드를 선정하고, 3) 기준이 되는 노드의 Radius와 레이블을 가져온다. 4) 기준이 되는 노드의 오른쪽에 삽입되면 Radisu를 증가시키고 왼쪽에 삽입되면 Radius를 감소시킨다. 5) 삽입되는 노드 / 서브트리에 대한 셀프 레이블을 계산하여 붙여준다.

소수를 표현할 수 있는 변수 Radius를 사용함으로써 XML 문서에 새로운 데이터의 삽입 시 전체 레이블에 대한 갱신 없이 삽입되는 데이터에 대해서만 레이블링을 해주면 된다. 또한 같은 위치에 지속적으로 새로운 데이터가 삽입되더라도 고정된 길이의 레이블을 유지할 수 있기 때문에 전체 레이블 저장공간의 효율을 얻을 수 있다.

4. 결론

인터넷과 유비쿼터스 환경에서 XML 데이터는 데이터의 교환과 저장 등의 표준으로 광범위하게 사용되고 있다. XML 데이터를 효과적으로 이용하는 방법으로 레이블링 방법이 연구되고 있지만 기존 연구들에서는 XML 문서에 대한 큰 레이블 저장공간이 필요하다는 문제점을 갖는다. 또한, 동적 XML 환경에 대한 고려로 새로운 데이터가 삽입될 경우 전체 레이블에 대한 변경 없이 삽입되는 데이터에 대한 레이블링만 하는 방법들이 제시되고 있지만, 레이블의 길이를 증가시켜 저장공간의 비효율성을 야기한다. 이 논문에서는 XML 문서를 원으로 표현하였다는 점과 매우 큰 XML 문서에 대한 전체 레이블 사이즈의 절감에 대한 고려를 했다는 점, 동적 XML 환경에 효율적인 레이블링 방법을 제시했다는 점에 의미를 둘 수 있다. 원으로 표현함으로써 회전 수(Rotation number)와 루트원(Root Circle) / 확장원(Extension Circle) 개념을 통해 매우 큰 XML 문서에 대한 전체 레이블 저장공간에 대한 효율을 얻었다. 또한 반지름(소수 표현 가능) 개념을 적용하여 레이블의 길이를 증가시키지 않고 삽입할 수 있어 레이블 저장공간에 효율적이며 삽입되는 데이터들의 레이블 간 충돌이 발생하지 않는다.

5. 참고문헌

[1] X.Wu, M.L.Lee, W.Hsu, “A Prime number Labeling Scheme for Dynamic Ordered XML Trees”, In Proc of the 20th International Conference on Data Engineering, 2004, pp 66-78.

[2] P.O’Neil, E.O’Neil, S.Pal, I.Cseri, G.Schaller, and N.Westbury, “ORDPATH: Insert-Friendly XML Node Labels”, In Proc of the 2004 ACM SIGMOD International Conference on Management of data, 2004, pp 903-908.

[3] M.Duong and Y.Zhang, “LSDX:A new Labeling Scheme for Dynamically Updating XML Data”, In Proc. Of the 16th Australian Database Conference, Newcastle,Australia,2005,pp.185-193.