



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2009-0004379
(43) 공개일자 2009년01월12일

(51) Int. Cl.

G06F 17/00 (2006.01) G06F 17/30 (2006.01)

(21) 출원번호 10-2007-0131067

(22) 출원일자 2007년12월14일

심사청구일자 2007년12월14일

(30) 우선권주장

1020060138740 2006년12월29일 대한민국(KR)

(71) 출원인

고려대학교 산학협력단

서울 성북구 안암동5가1 고려대학교 내

(72) 발명자

이상근

서울 동대문구 제기2동 153 안암골 벽산아파트
101동203호

고혜경

서울 노원구 월계2동 롯데캐슬루나 121동 1004호

박상현

서울 성북구 안암동2가 126-2번지 해승빌딩 305호

(74) 대리인

현중철

전체 청구항 수 : 총 19 항

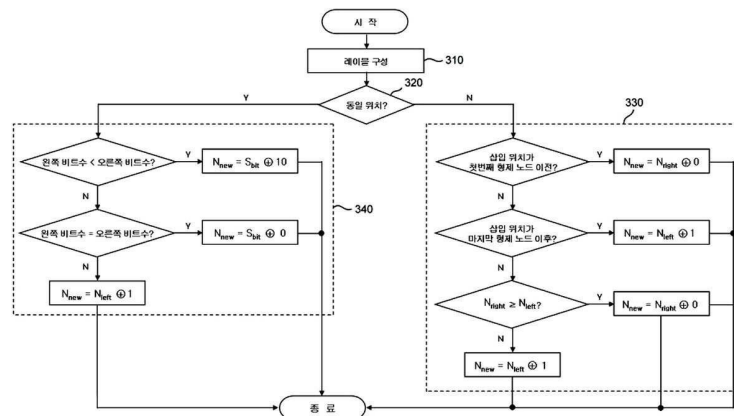
(54) XML 문서의 노드에 대한 레이블링 방법 및 장치

(57) 요약

본 발명은 XML 문서에서 노드의 갱신(신규 노드의 삽입)이 발생하였을 때의 노드에 대한 레이블링 방법에 관한 것이며, 아울러 삭제된 레이블을 재사용할 수 있는 레이블링 방법에 관한 것이다.

본 명세서에서 개시하는 레이블링 방법은 (a)확장성 표기 언어 문서에서 부모 노드를 상속하는 자식 노드의 레이블을 이전 스트링으로 구성하는 단계; (b)상기 확장성 표기 언어 문서에 새로운 노드가 삽입되는 경우, 상기 삽입되는 노드의 위치가 상기 문서로부터 삭제된 노드의 위치와 동일한지를 판단하는 단계; (c)상기 판단 결과 동일하지 아니한 경우에는 상기 삽입되는 위치와 이웃한 형제 노드의 레이블을 이용하여 상기 새로운 노드의 레이블을 설정하는 단계; 및 (d)상기 판단 결과 동일한 경우에는 상기 삭제된 노드의 레이블을 상기 새로운 노드의 레이블로 설정하는 단계를 포함하여 본 발명의 과제를 해결한다.

대표도



특허청구의 범위

청구항 1

- (a) XML 문서에서 부모 노드를 상속하는 자식 노드의 레이블을 이진 스트링으로 구성하는 단계;
- (b) 상기 XML 문서에 신규 노드가 삽입되는 경우, 상기 삽입되는 신규 노드의 위치가 상기 문서로부터 기 삭제된 노드의 위치와 동일한지를 판단하는 단계;
- (c) 상기 판단 결과 동일하지 아니한 경우에는 상기 삽입위치에 이웃한 형제 노드의 레이블을 이용하여 상기 신규 노드의 레이블을 설정하는 단계; 및
- (d) 상기 판단 결과 동일한 경우에는 상기 삭제된 노드의 레이블을 상기 신규 노드의 레이블로 설정하는 단계를 포함하는 XML 문서의 노드에 대한 레이블링 방법.

청구항 2

제 1 항에 있어서, 상기 (a)단계는

루트 노드는 비어 있는(null) 스트링으로 레이블링하여 이루어지는 것을 특징으로 하는 XML 문서의 노드에 대한 레이블링 방법.

청구항 3

제 1 항에 있어서, 상기 (a)단계는

상기 자식 노드의 레이블은 상기 부모 노드의 레이블을 프리픽스(prefix)로 하여 이루어지는 것을 특징으로 하는 XML 문서의 노드에 대한 레이블링 방법.

청구항 4

제 1 항에 있어서, 상기 (c)단계는

상기 삽입위치의 왼쪽 노드의 레이블의 이진 스트링(이하 N_{left})의 길이보다 오른쪽 노드의 레이블의 이진 스트링(이하 N_{right})의 길이가 크거나 같으면 상기 N_{right} 에 1 bit "0"을 연장시킨 레이블을 상기 신규 노드의 레이블로 설정하고,

상기 N_{left} 의 길이가 상기 N_{right} 의 길이 보다 크면 상기 N_{left} 에 1 bit "1"을 연장시킨 레이블을 상기 신규 노드의 레이블로 설정하여 이루어지는 것을 특징으로 하는 XML 문서의 노드에 대한 레이블링 방법.

청구항 5

제 1 항에 있어서, 상기 (c)단계는

첫 번째 형제 노드 이전에 상기 신규 노드가 삽입된 경우, 상기 첫 번째 형제 노드의 레이블의 이진 스트링의 최하위 비트(LSB) 다음에 1 bit "0"을 연장시킨 레이블을 상기 신규 노드의 레이블로 설정하여 이루어지는 것을 특징으로 하는 XML 문서의 노드에 대한 레이블링 방법.

청구항 6

제 1 항에 있어서, 상기 (c)단계는

마지막 형제 노드 이후에 상기 신규 노드가 삽입된 경우, 상기 마지막 형제 노드의 레이블의 이진 스트링의 최하위 비트(LSB) 다음에 1 bit "1"을 연장시킨 레이블을 상기 신규 노드의 레이블로 설정하여 이루어지는 것을 특징으로 하는 XML 문서의 노드에 대한 레이블링 방법.

청구항 7

제 4 항에 있어서,

상기 N_{left} 와 상기 N_{right} 의 길이를 비교하는 것은 레이블 스트링 오더의 사전적 순서를 비교하는 것으로서, 상기 두 개의 이진 스트링이 일치하면 스트링의 길이가 같은 것으로 판단하는 것을 특징으로 하는 XML 문서의 노드에 대한 레이블링 방법.

청구항 8

제 4 항에 있어서, 상기 N_{left} 와 N_{right} 의 길이를 비교하는 것은

상기 N_{left} 가 상기 N_{right} 의 프리픽스 이거나 상기 N_{left} 와 상기 N_{right} 를 왼쪽에서 오른쪽으로 사전적 (lexicographical) 비트 비교를 하여 상기 N_{left} 의 현재 비트가 "0"이고 상기 N_{right} 의 현재 비트가 "1"이면, 상기 N_{left} 보다 상기 N_{right} 의 스트링의 길이가 긴 것으로 판단하는 것을 특징으로 하는 XML 문서의 노드에 대한 레이블링 방법.

청구항 9

제 1 항에 있어서, 상기 (d)단계는

(d1)상기 신규 노드의 왼쪽 노드의 레이블의 이진 스트링의 비트수(이하 '왼쪽 비트수')와 오른쪽 노드의 레이블의 이진 스트링의 비트수(이하 '오른쪽 비트수')를 비교하는 단계;

(d2)상기 비교 결과 상기 오른쪽 비트수가 더 큰 경우에는, 상기 왼쪽 노드의 레이블의 이진 스트링과 상기 오른쪽 노드의 이진 스트링의 사전적 순서가 동일한 비트 스트링에 2 bit '10'을 이어붙이는(concatenate) 단계; 및

(d3)상기 비교 결과 상기 두 비트수가 같은 경우에는 상기 왼쪽 노드의 레이블의 이진 스트링과 상기 오른쪽 노드의 이진 스트링의 사전적 순서가 동일한 비트 스트링에 1 bit '0'을 이어붙이는 단계를 포함하여, 상기 이어 붙인 비트 스트링이 상기 신규 노드의 레이블로 설정되는 것을 특징으로 하는 XML 문서의 노드에 대한 레이블링 방법.

청구항 10

제 1 항 내지 제 9 항의 어느 한 항의 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록 매체.

청구항 11

XML 문서에서 부모 노드를 상속하는 자식 노드의 레이블을 이진 스트링으로 구성하는 레이블 구성부;

상기 XML 문서에 신규 노드가 삽입되는 경우, 상기 삽입되는 신규 노드의 위치가 상기 문서로부터 기 삭제된 노드의 위치와 동일한지를 판단하는 동일 위치 판단부; 및

상기 판단 결과 동일하지 아니한 경우에는 상기 삽입위치에 이웃한 형제 노드의 레이블을 이용하여 상기 신규 노드의 레이블을 설정하고, 동일한 경우에는 상기 삭제된 노드의 레이블을 상기 신규 노드의 레이블로 설정하는 레이블 설정부를 포함하는 XML 문서의 노드에 대한 레이블링 장치.

청구항 12

제 11 항에 있어서, 상기 레이블 구성부는

루트 노드는 비어 있는(null) 스트링으로 레이블링하는 것을 특징으로 하는 XML 문서의 노드에 대한 레이블링 장치.

청구항 13

제 11 항에 있어서, 상기 레이블 구성부는

상기 부모 노드의 레이블을 상기 자식 노드의 레이블의 프리픽스(prefix)로 지정하는 것을 특징으로 하는 XML 문서의 노드에 대한 레이블링 장치.

청구항 14

제 11 항에 있어서, 상기 동일 위치 판단부가 상기 새로이 삽입된 노드의 위치와 상기 삭제된 노드의 위치가 동일하지 아니하다고 판단한 경우 상기 레이블 설정부는

상기 삽입된 위치의 왼쪽 노드의 레이블의 이진 스트링(이하 N_{left})의 길이보다 오른쪽 노드의 레이블의 이진 스트링(이하 N_{right})의 길이가 크거나 같으면 상기 N_{right} 에 1 bit "0"을 연장시킨 레이블을 상기 신규 노드의 레이블로 설정하고,

상기 N_{left} 의 길이가 상기 N_{right} 의 길이 보다 크면 상기 N_{left} 에 1 bit "1"을 연장시킨 레이블을 상기 신규 노드의 레이블로 설정하는 것을 특징으로 하는 XML 문서의 노드에 대한 레이블링 장치.

청구항 15

제 11 항에 있어서, 상기 동일 위치 판단부가 상기 새로이 삽입된 노드의 위치와 상기 삭제된 노드의 위치가 동일하지 아니하다고 판단한 경우 상기 레이블 설정부는

첫 번째 형제 노드 이전에 상기 새로운 노드가 삽입된 경우, 상기 첫 번째 형제 노드의 레이블의 이진 스트링의 최하위 비트(LSB) 다음에 1 bit "0"을 연장시킨 레이블을 상기 신규 노드의 레이블로 설정하는 것을 특징으로 하는 XML 문서의 노드에 대한 레이블링 장치.

청구항 16

제 11 항에 있어서, 상기 동일 위치 판단부가 상기 새로이 삽입된 노드의 위치와 상기 삭제된 노드의 위치가 동일하지 아니하다고 판단한 경우 상기 레이블 설정부는

마지막 형제 노드 이후에 상기 신규 노드가 삽입된 경우, 상기 마지막 형제 노드의 레이블의 이진 스트링의 최하위 비트(LSB) 다음에 1 bit "1"을 연장시킨 레이블을 상기 신규 노드의 레이블로 설정하는 것을 특징으로 하는 XML 문서의 노드에 대한 레이블링 장치.

청구항 17

제 14 항에 있어서, 상기 레이블 설정부가 상기 N_{left} 와 상기 N_{right} 의 길이를 비교하는 것은

레이블 스트링 오더의 사전적 순서를 비교하는 것으로서, 상기 두 개의 이진 스트링이 일치하면 스트링의 길이가 같은 것으로 판단하는 것을 특징으로 하는 XML 문서의 노드에 대한 레이블링 장치.

청구항 18

제 14 항에 있어서, 상기 레이블 설정부가 상기 N_{left} 와 N_{right} 의 길이를 비교하는 것은

상기 N_{left} 가 상기 N_{right} 의 프리픽스 이거나 상기 N_{left} 와 상기 N_{right} 를 왼쪽에서 오른쪽으로 사전적(lexicographical) 비트 비교를 하여 상기 N_{left} 의 현재 비트가 "0"이고 상기 N_{right} 의 현재 비트가 "1"이면, 상기 N_{left} 보다 상기 N_{right} 의 스트링의 길이가 긴 것으로 판단하는 것을 특징으로 하는 XML 문서의 노드에 대한 레이블링 장치.

청구항 19

제 11 항에 있어서, 상기 동일 위치 판단부가 상기 새로이 삽입된 노드의 위치와 상기 삭제된 노드의 위치가 동일하다고 판단한 경우 상기 레이블 설정부는

상기 신규 노드의 왼쪽 노드의 레이블의 이진 스트링의 비트수(이하 '왼쪽 비트수')와 오른쪽 노드의 레이블의 이진 스트링의 비트수(이하 '오른쪽 비트수')를 비교하는 비트수 비교부; 및

상기 비교 결과 상기 오른쪽 비트수가 더 큰 경우에는 상기 왼쪽 노드의 레이블의 이진 스트링과 상기 오른쪽 노드의 이진 스트링의 사전적 순서가 동일한 비트 스트링에 2 bit '10'을, 상기 비교 결과 상기 두 비트수가 같은 경우에는 상기 사전적 순서가 동일한 비트 스트링에 1 bit '0'을 이어붙이는(concatenate) 재설정부를 포함하여,

상기 삭제된 노드의 레이블이 상기 신규 노드의 레이블로 설정(재사용)되는 것을 특징으로 하는 XML 문서의 노드에 대한 레이블링 장치.

명세서

발명의 상세한 설명

기술분야

- <1> 본 발명은 XML(eXtensible Markup Language) 문서의 노드에 대한 레이블링 방법 및 장치에 관한 것으로, 특히 확장성 표기 언어 문서에서 노드의 갱신(신규 노드의 삽입)이 발생하였을 때의 노드에 대한 레이블링 방법 및 그 장치에 관한 것이며, 아울러 삭제된 레이블을 재사용할 수 있는 레이블링 방법 및 장치에 관한 것이다.

배경기술

- <2> 최근 확장성 표기 언어(XML)를 이용해서 전자문서를 작성하는 예가 늘어나고 있다. 세계적으로 XML의 이용률이 급속히 높아지고 있으며, 국내에서도 일반 기업체는 물론, 정부와 관공서, 공공 기관에서의 XML 도입이 크게 늘고 있다. 이들 전자 문서들은 각 기관 내에서 정한 문법에 기반해서 작성되는 경우가 일반적이다.
- <3> XML로 작성된 전자문서들이 급속히 늘면서 서버에서 문서들을 유지하고 있다가 사용자 요청에 따라 원하는 전자문서를 찾아서 제공하는 클라이언트/서버 구조로 전자문서가 관리되는 예가 늘어나고 있다. 이러한 클라이언트/서버 구조는 도 1a에 나타낸 바와 같다. 도 1a는 일반적인 확장성 표기 언어 기반의 전자문서를 요청/제공하는 클라이언트/서버 구조를 개략적으로 나타낸 도면이다.
- <4> 이때, 클라이언트는 문서를 요청하고 사용하는 측을 나타내며, 서버는 문서 제공자를 나타낸다. 따라서 문서를 요청하고 사용하는 시스템이 문서를 저장하고 제공할 수 있는 기능을 동시에 갖는 경우, 역할에 따라 클라이언트가 동시에 서버가 될 수 있다.
- <5> 이와 같은 XML 문서들을 검색하는 언어로는 XQL(XML Query Language), XQuery 등의 언어들이 있다. 이렇게 요청되어 제공되는 문서를 사용하는 경우에 있어, 변경되거나 추가된 내용들만을 알려 주기 위하여 이를 별도로 표현할 방법이 없어 최초 제공 당시에 이미 제공된 내용을 포함하는 전체 내용을 모두 반복하여 전송하여야 한다.
- <6> 최근의 연구에서 제안된 Prime 방법이 있다. Prime 방법은 문서의 순서를 위해 Chinese Remainder Theorem(CRT)를 이용하고 각각의 XML 노드에 대해서 CRT 내의 SC(Simultaneous Congruence) 값을 계산해야 하기 때문에 XML tree 내에 새로운 노드가 삽입 시, 재레이블링 대신에 노드의 새로운 순서를 위한 SC 값을 재계산하게 된다.
- <7> 따라서, 종래의 레이블링 방법은 새로운 노드가 삽입 시 XML 문서 전체를 새로 재레이블링을 하기 때문에 갱신 비용이 높고, Prime 방법의 경우에는 새로운 노드들이 XML tree에 삽입될 때 재레이블링은 하지 않지만, CRT내의 SC 값을 다시 계산해야 하기 때문에, 재계산 비용이 높아지고 재계산을 위한 처리 시간이 매우 길어지는 문제점이 있다.

발명의 내용

해결하고자 하는 과제

- <8> 따라서, 본 발명이 해결하고자 하는 첫 번째 과제는 기존의 어느 노드들도 재레이블링하지 않고 XML 트리 내에 순서에 민감한 노드가 삽입되었을 때 어느 값도 재계산할 필요가 없으며, 순서가 있는 XPath/XQuery의 빠른 질의 처리를 제공하고, 순서가 있는 XML의 갱신 비용을 절감할 수 있는 확장성 표기 언어 문서의 갱신을 위한 레이블링 방법을 제공하는데 있다.
- <9> 본 발명이 해결하고자 하는 두 번째 과제는 상기의 확장성 표기 언어 문서의 갱신을 위한 레이블링 방법이 적용되는 확장성 표기 언어 문서의 갱신을 위한 레이블링 장치를 제공하는데 있다.
- <10> 본 발명이 해결하고자 하는 세 번째 과제는 XML 트리 내에 순서에 민감한 노드가 새로 삽입되었을 때 소정의 경우 그 삽입된 노드의 레이블에 상기 XML 트리에 존재하다가 삭제된 레이블을 설정하여 레이블의 길이와 저장 공

간을 줄일 수 있는 효율적인 레이블링 방법을 제공하는데 있다.

<11> 본 발명이 해결하고자 하는 네 번째 과제는 XML 트리 내에 순서에 민감한 노드가 새로 삽입되었을 때 소정의 경우 그 삽입된 노드의 레이블에 상기 XML 트리내 존재하다가 삭제된 레이블을 설정하여 레이블의 길이와 저장 공간을 줄일 수 있는 효율적인 레이블링 장치를 제공하는데 있다.

<12> 본 발명이 해결하고자 하는 다섯 번째 과제는 상기의 확장성 표기 언어 문서의 갱신을 위한 레이블링 방법과 삭제된 레이블을 재사용할 수 있는 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공하는데 있다.

과제 해결수단

<13> 상기의 과제들을 해결하기 위하여, 본 발명은 (a)XML 문서에서 부모 노드를 상속하는 자식 노드의 레이블을 이진 스트링으로 구성하는 단계; (b)상기 XML 문서에 신규 노드가 삽입되는 경우, 상기 삽입되는 신규 노드의 위치가 상기 문서로부터 기 삭제된 노드의 위치와 동일한지를 판단하는 단계; (c)상기 판단 결과 동일하지 아니한 경우에는 상기 삽입위치에 이웃한 형제 노드의 레이블을 이용하여 상기 신규 노드의 레이블을 설정하는 단계; 및 (d)상기 판단 결과 동일한 경우에는 상기 삭제된 노드의 레이블을 상기 신규 노드의 레이블로 설정하는 단계를 포함하는 XML 문서의 노드에 대한 레이블링 방법을 제공한다.

<14> 상기의 과제들을 해결하기 위하여, 본 발명은 확장성 표기 언어 문서에서 부모 노드를 상속하는 자식 노드의 레이블을 이진 스트링으로 구성하는 레이블 구성부; 상기 확장성 표기 언어 문서에 신규 노드가 삽입되는 경우, 상기 삽입되는 신규 노드의 위치가 상기 문서로부터 기 삭제된 노드의 위치와 동일한지를 판단하는 동일 위치 판단부; 및 상기 판단 결과 동일하지 아니한 경우에는 상기 삽입위치에 이웃한 형제 노드의 레이블을 이용하여 상기 새로운 노드의 레이블을 설정하고, 동일한 경우에는 상기 삭제된 노드의 레이블을 상기 신규 노드의 레이블로 설정하는 레이블 설정부를 포함하는 XML 문서의 노드에 대한 레이블링 장치를 제공한다.

<15> 상기의 과제들을 해결하기 위하여, 본 발명은 상기의 확장성 표기 언어 문서의 갱신을 위한 레이블링 방법 및 삭제된 레이블을 재사용할 수 있는 방법을 컴퓨터에서 실행시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공한다.

효과

<16> 본 발명에 의하면, 이진 스트링(binary string)의 특성을 이용한 새로운 프리픽스 기반 레이블링을 구현함으로써, 기존의 어느 노드들도 재레이블링하지 않고 XML 트리 내에 순서에 민감한 노드가 삽입되었을 때 어느 값도 재계산할 필요가 없으며, 순서가 있는 XPath/XQuery의 빠른 질의 처리를 제공하고, 순서가 있는 XML의 갱신 비용을 절감할 수 있는 효과가 있다.

<17> 아울러 삭제된 레이블의 재사용이 가능하기 때문에 레이블의 사이즈를 줄일 수 있어 질의 처리 시간을 절감할 수 있고 레이블의 저장 공간을 절약할 수 있는 효과도 갖는다.

발명의 실시를 위한 구체적인 내용

<18> 이하, 본 발명의 기술적 과제의 해결 방안을 명확화하기 위한 발명의 구성을 본 발명의 바람직한 실시예에 근거하여 첨부 도면을 참조하여 상세히 설명하되, 도면의 구성요소들에 참조번호를 부여함에 있어서 동일 구성요소에 대해서는 비록 다른 도면상에 있더라도 동일 참조번호를 부여하였으며 당해 도면에 대한 설명시 필요한 경우 다른 도면의 구성요소를 인용할 수 있음을 미리 밝혀둔다.

<19> 노드 사이 관계(relationship)의 결정을 쉽게 파악하기 위해서 XML 트리 내에 노드들은 전형적으로 두 노드 사이의 조상-후손관계들을 레이블링 한다. 즉, 간단하고 좋은 레이블링 방법은 XML 질의를 효율적으로 처리하기 위해서 꼭 필요하다. XML 문서는 웹(web) 상에서 빈번하게 변하고, 새로운 노드가 삽입될 때 높은 문서 갱신 비용이 발생한다.

<20> 본 발명은 웹상에서 빈번한 XML 문서의 동적인 갱신 시 레이블 갱신 비용을 줄이고 순서에 민감(order-sensitive)한 질의를 효율적으로 처리하는 기술에 관한 것이다. 본 발명은 XML 문서의 동적인 갱신을 지원하는 기술에 관한 것으로, 특히 레이블링 방법을 이용하여 XML 트리의 엘리먼트 관계 (조상-후손, 부모-자식, 형제)를 쉽게 파악할 수 있어 빠른 질의 처리를 제공하고, XML 문서의 빈번한 갱신 시 재레이블링이 필요 없는 효율적인 레이블링 방법에 관한 것이다.

- <21> 아울러 XML 문서에서 삭제된 노드가 있을 경우, 그 삭제된 노드와 동일한 위치에 새로운 노드가 삽입되었을 때 삽입된 노드에 대한 레이블을 상기 삭제된 노드의 레이블로 설정하여 이용할 수 있는 레이블링 방법에 관한 것이다.
- <22> 본 발명은 이진 스트링(binary string)의 특징을 이용한 프리픽스(Prefix) 기반 이진 스트링 방법을 이용하여, 각각의 이진 스트링의 문자를 단위비트(예를 들어, 1 비트)로 저장하고 자식 노드는 부모 노드를 상속 받아 레이블을 구성하는 과정과 XML 문서의 갱신 시 새로운 노드를 삽입하기 위해 사전적 순서(lexicographical order)를 갖는 4가지 경우의 스트링 오더를 이용하여 왼쪽 스트링의 길이와 오른쪽 스트링의 길이를 비교한 뒤, 스트링 1이나 0을 본래의 스트링에 연결시키는 과정을 포함한다.
- <23> 이를 통해 순서에 민감한 노드가 삽입 시에 다른 노드들에 대한 재레이블링 필요 없이 스트링의 길이 비교 후 새로운 노드에만 레이블링을 하기 때문에 갱신 비용이 매우 감소하고, 자식 노드가 부모 노드를 상속 받기 때문에 순서에 민감한 질의 처리 시 수행 시간을 감소시키는 효과를 얻을 수 있다.
- <24> 아울러 노드의 삭제가 있고, 그 후에 삭제된 노드의 위치에 새로운 노드가 삽입되는 경우에는 삭제된 레이블의 재사용이 가능하기 때문에 레이블의 사이즈를 줄일 수 있어 질의 처리 시간을 절감할 수 있고 레이블의 저장 공간을 절약할 수 있는 효과도 얻을 수 있다.
- <25> 도 1b는 순서가 있는 XML 데이터의 구조를 트리 형태로 나타낸 것이다.
- <26> XPath와 XQuery는 주요 XML 질의 언어로 선형 경로나 간가지 형태로써 XML 문서의 구조를 표현한다.
- <27> 예를 들면, XPath 질의 : /book[/title]//section[2]/preceding-sibling::section은 section[2]의 형제(sibling)들 중 모든 section 노드들을 찾고, section sibling 노드들은 section[2]의 전에 나타내야 한다. 다시 말하면, section[2]는 book("/")의 자손이어야 한다.
- <28> 위의 질의는 선형 경로나 간가지 형태에서는 문제가 되지 않지만, 주요 작용은 조상-후손(ancestor-descendant), 부모-자식(parent-child), 형제(sibling)와 순서 관계를 효율적으로 결정하는 것이다.
- <29> 만일 XML이 정적이라면, 기존의 레이블링 방법들은 다른 질의들을 효율적으로 처리할 수 있지만, XML이 동적이라면, 효율적으로 갱신을 처리하는 연구들은 중요한 주제가 된다. 도 1b에서와 같이 XML 트리 내의 엘리먼트(노드)들은 본질적으로 순서를 갖는다. 즉, XML 문서 내의 엘리먼트 순서는 문서 순서(document order)로 언급한다.
- <30> 도 2는 본 발명의 일 실시 예에 따른 XML 문서의 노드에 대한 레이블링 장치의 구성을 제시한 도면이며, 도 3은 본 발명의 일 실시 예에 따른 XML 문서의 노드에 대한 레이블링 방법의 흐름을 제시한 도면이다.
- <31> 도 2의 확장성 표기 언어 문서의 갱신을 위한 레이블링 장치는 하나 이상의 마이크로 프로세서, 마이크로 프로세서에서 액세스가 가능한 메모리 및 메모리로부터 데이터를 기록/독출 하기 위한 수단을 포함할 수 있다.
- <32> 이하에서 설명한 각각의 구성 요소는 관련된 연산을 수행하는 프로세서 기타 하드웨어적인 수단으로 구현될 수 있다. 한편, 각각의 구성 요소는 관련된 연산을 수행하는 프로그램의 세그먼트들로 구현될 수도 있다.
- <33> 레이블 구성부(210)는 XML 문서에서 부모 노드를 상속하는 자식 노드의 레이블을 이진 스트링으로 구성한다(310 과정). 레이블 구성부(210)는 바람직하게 루트 노드는 비어 있는(null) 스트링으로 레이블링하고, 부모 노드의 레이블을 상기 자식 노드의 레이블의 프리픽스(prefix)로 하여 상속시킬 수 있다.
- <34> 동일 위치 판단부(220)는 XML 문서에 신규 노드가 삽입되는 경우, 삽입되는 신규 노드의 위치가 XML 문서에 존재하였다가 삭제된 노드의 위치와 동일한지를 판단한다(320 과정). 동일 위치 판단부(220)의 의의 내지 필요성은 후술될 것이다.
- <35> 레이블 설정부(230)는 상기 판단 결과 동일하지 아니한 경우에는 신규 노드의 삽입위치에 이웃한 형제 노드의 레이블을 이용하여 신규 노드의 레이블을 설정하고(330 과정), 동일한 경우에는 삭제된 노드의 레이블을 신규 노드의 레이블로 설정한다(340 과정).
- <36> 먼저 상기 판단 결과 동일하지 아니한 경우에 있어서의 레이블 설정 방안(330 과정)을 제시한다.
- <37> 레이블 설정부(230)는 우선 신규 노드가 삽입된 위치의 왼쪽 노드의 레이블의 이진 스트링(이하 N_{left})의 길이와 오른쪽 노드의 레이블의 이진 스트링(이하 N_{right})의 길이를 비교한다.

- <38> 비교 결과 N_{right} 가 크거나 같으면 N_{right} 에 1 bit "0"을 연장시킨 레이블을 상기 신규 노드의 레이블로 설정하며, N_{left} 의 길이가 N_{right} 의 길이 보다 크면 N_{left} 에 1 bit "1"을 연장시킨 레이블을 신규 노드의 레이블(N_{new})로 설정한다.
- <39> 만일 신규 노드가 첫 번째 형제 노드 이전에 삽입되는 경우(도 4의 a 또는 d 노드의 경우)에는 왼쪽 노드에 해당되는 노드가 없으므로, 첫 번째 형제 노드의 레이블의 이진 스트링의 최하위 비트(LSB) 다음에 1 bit "0"을 연장시킨 레이블을 상기 신규 노드의 레이블로 설정하게 된다.
- <40> 이에 대응하여 만일, 신규 노드가 마지막 형제 노드 이후에 삽입되는 경우(도 4의 c 또는 f 노드의 경우)에는 오른쪽 노드에 해당되는 노드가 없으므로, 마지막 형제 노드의 레이블의 이진 스트링의 최하위 비트(LSB) 다음에 1 bit "1"을 연장시킨 레이블을 상기 신규 노드의 레이블로 설정하게 된다.
- <41> 한편 레이블 설정부(230)가 상기 판단 결과 동일하지 아니한 경우에 있어서 레이블을 설정하는 경우에 N_{left} 와 상기 N_{right} 의 길이를 비교하는 것은 레이블 스트링 오더의 사전적 순서(lexicographical order)를 비교하는 것으로서, 상기 두 개의 이진 스트링이 일치하면 스트링의 길이가 같은 것으로 판단하는 것이다.
- <42> 구체적으로는 N_{left} 가 N_{right} 의 프리픽스(prefix) 이거나 N_{left} 와 N_{right} 를 왼쪽에서 오른쪽으로 사전적으로 비트 대 비트(bit by bit) 비교를 하여 N_{left} 의 현재 비트가 "0"이고 N_{right} 의 현재 비트가 "1"이면, N_{left} 보다 N_{right} 의 스트링의 길이가 긴 것으로 판단하는 것이다. 이와 같이 사전적 비트 비교는 두 개 이진 스트링의 MSB(Most Significant Bit)에서 LSB(Least Significant Bit) 방향으로 비트 대 비트(bit by bit) 비교를 하는 것이다.
- <43> 도 4는 상기한 레이블링 기법의 예를 제시한 도면이다.
- <44> XML 트리 내의 노드의 레이블은 루트(root) 노드는 빈(null) 값으로 남겨두고, 루트 노드의 첫 번째 자식부터 레이블링을 하게 된다. 노드의 레이블을 N으로 표현했을 때, N의 첫 번째 자식(child)은 $label(N). "10"$, N의 두 번째 자식은 $label(N). "110"$, 그리고 N의 i번째 자식은 $label(N). "1^i 0"$ 으로 레이블링 된다. 레이블링 값은 모두 이진 스트링을 이용한다.
- <45> 예를 들면, 도 4에서 루트(root) 노드는 빈(null) 스트링으로 레이블링 되고, 루트 노드가 빈 스트링이기 때문에 부모의 스트링을 물려받는 네 개의 자식 노드들의 prefix_label은 모든 빈 스트링들이다. 따라서 첫 번째 자식 노드의 self_label은 "10", 두 번째 자식 노드의 self_label은 "110", 세 번째 자식 노드는 "1110", 네 번째 자식 노드는 "11110"을 갖는다.
- <46> 도 4에서 노드의 레이블은 부모의 레이블과 자신의 레이블을 연결한 것이다. 즉, 두 노드 "10"과 "10.110"에서 만일 "10"이 "10.110"의 prefix이면, "10"은 "10.110"의 부모(parent)이다.
- <47> 도 4에서 순서가 있는 XML 트리 내의 노드들을 삽입하기 위해서 삽입 스트링 오더(Insert string order)를 이용한다. 삽입 노드의 왼쪽과 오른쪽에 인접한 두 노드의 레이블의 이진 스트링을 각각 N_{left} 와 N_{right} 라고 할 때, 새로 삽입되는 노드의 레이블 N_{new} 는 항상 처음 할당된 노드들의 수정 없이 삽입 스트링 오더에 의해서 할당된다.
- <48> 삽입 스트링 오더는 다음의 네 가지 경우와 같다. 여기서 $len(N)$ 은 상기 언급한 사전적 의미의 비트(bit) 길이로 정의된다. 이때, " $\oplus$ "는 연결(concatenation)을 의미한다.
- <49> 1. $len(N_{left}) \leq len(N_{right})$ then, $N_{new} = N_{right} \oplus 0$
- <50> 2. $len(N_{left}) > len(N_{right})$ then, $N_{new} = N_{left} \oplus 1$
- <51> 3. 첫 번째 형제 노드 전에 새로운 노드가 삽입되는 경우, $N_{new} = N_{left} \oplus 0$
- <52> 4. 마지막 형제 노드 후에 새로운 노드가 삽입되는 경우, $N_{new} = N_{right} \oplus 1$
- <53> 다음의 예는 삽입 스트링 오더의 위 네 가지 경우에 대한 처리 방법을 나타낸 것이다.
- <54> 첫째, 두 노드 사이에 새로운 노드가 삽입 될 경우는 왼쪽 노드의 self_label의 길이보다 오른쪽 노드의 self_label의 길이가 크거나 같은 경우, 새로 삽입된 노드의 self_label은 오른쪽 노드의 self_label에 1 bit

"0"을 연결시킨다. 다른 경우 왼쪽 노드의 self_label의 길이가 오른쪽 노드의 self_label의 길이 보다 큰 경우, 새로 삽입된 노드의 self_label은 왼쪽 노드의 self_label에 1 bit "1"을 연결시킨다.

<55> 둘째, 첫 번째 형제 노드 전에 새로운 노드가 삽입 될 경우는 새로 삽입된 노드의 self_label은 첫 번째 self_label의 마지막 문자에 1 bit "0"을 연결시킨다.

<56> 셋째, 마지막 형제 노드 후에 새로운 노드가 삽입될 경우, 새로 삽입된 노드 self_label은 마지막 형제 노드의 self_label에 1 bit "1"을 연결시킨다.

<57> 본 발명에서 레이블링의 삽입 스트링 오더는 숫자상의 순서가 아닌 사전적 순서(lexicographical order)를 갖는다. 한편, 상기 이진 스트링의 사전적 순서를 위한 정의를 설명하면 다음과 같다.

<58> 본 설명에서 주어진 두 개의 이진 스트링을 N_{left} 와 N_{right} 라고 할 때, 만일 두 개의 스트링이 정확하게 같다면 사전적으로 동등하다고 한다. 만일 N_{left} 가 N_{right} 의 prefix이거나, N_{left} 와 N_{right} 를 왼쪽에서 오른쪽으로 사전적 비트 비교를 할 경우, N_{left} 의 현재 비트가 "0"이고 N_{right} 의 현재 비트가 "1"일 때, $N_{left} < N_{right}$ 이고 비트 오더 비교를 멈추게 된다.

<59> 이하에서는 두 개의 이진 스트링들을 상술한 방법을 이용하여 스트링 오더를 비교하는 예를 보여준다. 주어진 두 개의 이진 스트링을 "10"과 "110"라고 할 때, 두 스트링을 왼쪽에서 오른쪽으로 비교 할 경우 "10"의 두 번째 비트는 "0"이고, "110"의 두 번째 비트는 "1"이기 때문에 스트링 오더는 "10" < "110"이 된다.

<60> 본 설명에서는 도 4에 나타낸 6가지 경우로 노드가 삽입되는 것을 예로 하여 설명한다. 도 4의 노드 "a"가 삽입될 경우, "a"의 self_label은 "100"이 된다("10" → "100").

<61> 노드 "b"가 삽입될 경우, "b"의 왼쪽 self_label은 길이 2의 "10"이고 오른쪽 self_label은 길이 3의 "110"이기 때문에 삽입된 노드 "b"는 오른쪽 self_label에 "0"을 연결시킨다("110" $\oplus$ "0" → "1100"). 노드 "c"가 삽입될 경우 "c"의 self_label은 "111101"이 된다("11110" $\oplus$ "1" → "111101").

<62> 위의 예와 같이 레이블의 스트링 크기를 비교하여 삽입 시 새로운 노드만을 레이블링을 해주기 때문에 동적인 문서 갱신 시 비용이 절감된다.

표 1

<63>

입력 : XML 문서
출력 : 각 노드의 레이블
Description
1. for XML 문서의 각각의 노드의 모든 형제 노드에 대하여
2. for 첫 번째 형제 노드, $selflabel[1] = "10"$ 에 대하여
3. if 형제 노드들의 수 $Snum > 1$ 경우
$selflabel[Snum] = "110";$
$selflabel = labeling("1" selflabel)$
/* 는 연결(concatenation)을 의미 */
4. $label = prefix_label\ delimiter\ self_label$ 의 각각의 엘리먼트

<64> 표 1에 기재된 알고리즘 1은 XML 문서의 레이블링 방법을 기술한 것이다. 알고리즘 1을 이용하여 XML 문서에 대하여 Prefix 기반 이진 스트링 레이블링 방법을 실행한다. 레이블의 초기값은 루트의 첫 번째 자식의 self_label인 "10"을 갖는다. 형제 노드들의 수(Snum)가 하나 이상일 경우는 계속해서 앞 형제의 self_label에 스트링 "1"을 연결(concatenation)하여 새로운 레이블을 생성한다. 자기 자신의 완전한 레이블링은 부모에게 물려받은 prefix_label과 자기 자신의 레이블인 self_label을 연결시켜서 최종 값을 갖는다. 이때, delimiter는 구분 기호를 의미한다.

표 2

<65>

입력 : $selflabel_{left}$, $selflabel_{right}$

출력 : 새로 삽입된 노드의 레이블 $selflabel_{new}$

Description

1. 인접한 노드 $selflabel_{left}$ 와 $selflabel_{right}$ 의 크기 값 생성;
2. if size ($selflabel_{left}$) $\leq$ size ($selflabel_{right}$) 일 경우
 $selflabel_{new} = selflabel_{right}$ "0";
3. else if
 $selflabel_{new} = selflabel_{left}$ "1";

<66>

표 2에 기재된 알고리즘 2는 새로운 노드 삽입 시 레이블링을 갱신하는 방법을 기술한 것이다. 알고리즘 2는 사전적 순서를 갖는 이진 스트링 사이에 항상 새로운 스트링을 삽입할 수 있는 방법을 제공한다. 알고리즘 2의 수행 절차는 다음과 같다. 첫째, 노드의 모든 형제 자식 노드들에 상기에 기술한 경우에 따라서 새로운 self_label을 할당한다.

<67>

다음으로 신규 삽입 노드의 위치가 삭제된 노드의 위치와 동일한 경우에 있어서의 레이블 설정 방안(340 과정)을 제시한다.

<68>

본 발명은 삭제된 레이블을 재사용할 수 있는 방안도 함께 제시한다고 위에서 언급하였다. 본 발명에 의한 레이블링 기법은 XML 문서에 새로운 노드가 삽입될 때, 새롭게 삽입된 노드에 대해서만 레이블을 부여하므로 상술한 바와 같이 다른 노드에 대해서는 재 레이블링(relabeling)을 할 필요가 없는 장점이 있다.

<69>

그러나 새로 삽입되는 노드에 대한 레이블을 부여할 때 마다 레이블의 비트 수가 증가하기 때문에 레이블의 저장 공간이 늘어날 우려가 있다. 특히 노드가 삭제되고 삭제된 노드와 동일한 위치에 다른 노드가 신규로 삽입되는 경우에는 상기한 기법에 의해서만 삭제된 노드의 레이블과는 다른 레이블이 부여되므로 이러한 우려는 더 더욱 심각해진다.

<70>

따라서 이러한 우려를 해소하기 위하여 상기한 기법의 보완이 요구되는 바, 본 발명은 삭제된 노드를 재사용할 수 있는 기법도 아울러 제시하는 것이다.

<71>

상기한 우려는 도 4를 참고하여 다음의 예를 들어보면 쉽게 파악할 수 있다.

<72>

예를 들어, 레이블 “100”과 “110” 사이의 레이블 “10”이 삭제되었을 때, 동일한 위치에 새로이 삽입되는 노드의 레이블은 본 발명에 의한 상기한 레이블링 기법에 의하면 “10”이 아닌 “1100”이 된다. 상기 삭제된 레이블 “10”은 레이블 “100”과 “110”보다 작은 길이를 갖기 때문이다. 따라서 삽입된 위치가 동일한 위치임에도 불구하고 레이블이 달리 부여되며 아울러 레이블의 비트수가 증가되는 것을 알 수 있다.

<73>

이러한 우려를 해결하기 위해 본 발명은 삭제된 레이블을 재사용할 수 있는 기법을 제안한다.

<74>

삭제된 레이블의 재사용은 삭제된 노드의 위치에 또 다른 새로운 노드가 삽입되는 경우, 그 노드에 대해서는 별개의 레이블을 부여할 필요가 없다는 것을 그 출발 개념으로 한다. 이러한 출발 개념에 착안하여 동일 위치 판단부(220)에 의해 신규 삽입 노드의 위치가 판단될 필요가 있는 것이다.

<75>

삭제된 노드의 레이블을 삭제된 노드와 동일한 위치에 신규로 삽입되는 노드의 레이블로 재사용할 수 있도록 하는 본 발명에 의한 기법은 다음과 같다.

<76>

동일 위치 판단부(220)에 의해 신규 삽입 노드의 위치가 삭제된 노드의 위치와 같다고 판명되는 경우, 레이블 설정부(230)는 삭제된 노드의 레이블을 그대로 신규 노드의 레이블(N_{new})로 설정하게 되는데 그 설정의 바람직한 기법은 다음과 같다.

<77>

본 발명은 삭제된 노드의 레이블을 신규 노드의 레이블로 설정(재사용)함에 있어서는 도 2 내지 도 3에 제시된 레이블링 기법을 역으로 이용한다. 즉, 도 2 내지 도 3에 제시된 레이블링 기법은 결국 신규 노드에 대한 레이블을 부여할 때 신규 노드의 이웃한 노드(형제 노드)의 레이블에 비트 스트링을 이어 붙여 부여하는 것이므로, 역으로 삭제된 노드의 레이블을 재사용하게 하는 것도 새로이 삽입되는 노드에 이웃하는 노드와 공통되는 비트

가 많다는 점에 착안하는 것이다.

- <78> 비트수 비교부(2301)는 신규 삽입 노드의 왼쪽 노드의 레이블의 이진 스트링의 비트수(이하 ‘왼쪽 비트수’)와 오른쪽 노드의 레이블의 이진 스트링의 비트수(이하 ‘오른쪽 비트수’)를 비교한다. 여기서 비트수는 이진 스트링의 비트의 길이를 의미하는 것으로 도 2 내지 도 3에 제시된 길이와는 다른 순수한 물리적 개념의 길이이다.
- <79> 제설정부(2302)는 오른쪽 비트수가 더 큰 경우에는 신규 노드의 왼쪽 노드의 레이블의 이진 스트링과 오른쪽 노드의 이진 스트링의 사전적 순서가 동일한 비트 스트링(S_{bit})에 2 bit '10'을, 두 비트수가 같은 경우에는 상기 사전적 순서가 동일한 비트 스트링(S_{bit})에 1 bit '0'을 이어 붙여(concatenate) 삭제된 노드의 레이블을 그대로 신규 노드의 레이블로 재사용할 수 있도록 한다.
- <80> 이를 요약하면 다음과 같이 기술할 수 있다.
- <81> 1) 왼쪽 비트수 < 오른쪽 비트수; $N_{new} = S_{bit} \oplus 10$
- <82> 2) 왼쪽 비트수 = 오른쪽 비트수; $N_{new} = S_{bit} \oplus 0$
- <83> 여기서 $\oplus$ 는 도 2 내지 도 3에서의 경우와 마찬가지로 이어붙이기(concatenation)를 의미한다.
- <84> 삭제된 노드의 레이블을 삭제된 노드와 동일한 위치에 신규 삽입되는 노드의 레이블로 재사용할 수 있는 사실을 예를 들어 보면 다음과 같다.
- <85> 도 4에서 레이블 “100”과 “110” 사이의 “10”이 삭제되고, 삭제된 위치에 신규한 노드가 삽입했다고 가정한다.
- <86> 이 경우 상기한 S_{bit} 는 사전적 순서가 동일한 비트 스트링을 의미하므로 본 예에 있어서 S_{bit} 는 ‘1’이 된다. 한편 “100”과 “110”의 비트수는 동일하므로 상기한 조건 2)에 의해 $N_{new} = S_{bit} \oplus 0 = “1” \oplus “0” = “10”$ 이 된다. 즉, 삭제된 노드의 레이블 “10”이 신규 삽입되는 노드의 레이블로 설정(재사용)되는 것을 확인할 수 있다.
- <87> 본 발명의 성능 평가를 통한 실험 측정값은 도 5 내지 도 7b를 통해 나타난다.
- <88> 성능 평가는 기존에 제안된 방법 DeweyID, Interval, Prime과 본 발명을 저장용량(storage), 질의(query), 그리고 갱신(update)의 세 방법으로 평가하였다. 평가를 위해서 Niagara Project에서 제공하는 실세계(real-world) XML 데이터를 이용하였다.
- <89> 도 5는 본 발명을 통해 얻어지는 질의 처리 시간(응답 시간)을 나타내는 도식화된 그래프이다.

표 3

Query		Max. # of nodes
Q1	/play//act[4]	185
Q2	/play//act[3]//following::act	370
Q3	/play//act/personal	969
Q4	/speech[4]//preceding::line	66,946
Q5	/act//following-sibling::speech[3]	143,725
Q6	/play/speech	154,755

- <91> 질의 처리 시간 측정을 위해 사용한 테스트 질의는 표 3와 같다. 표 3은 측정된 데이터셋으로부터 반환된 노드의 수와 순서가 있는 질의와 순서가 없는 질의를 나타낸다. 도 5에서 Q3와 Q6(순서가 없는 질의)을 제외하고, 발명된 방법은 다른 4개(Q1, Q2, Q4, Q5)의 순서가 있는 질의 처리에서 매우 빠른 성능을 보인다.
- <92> 도 6a 및 도 6b는 본 발명의 성능 평가 중 각각 재레이블링 되는 노드의 수와 재레이블링 처리 시간을 나타낸 그래프이다. 엘리먼트(노드)들은 순서에 민감하며, 성능 평가를 위해 6,636개의 노드를 가지는 XML 문서의 갱신 성능을 측정하였다. 도 6a 및 도 6b는 본 발명과 다른 방법들 간의 성능 평가를 보여주는 도식화된 그래프이다.

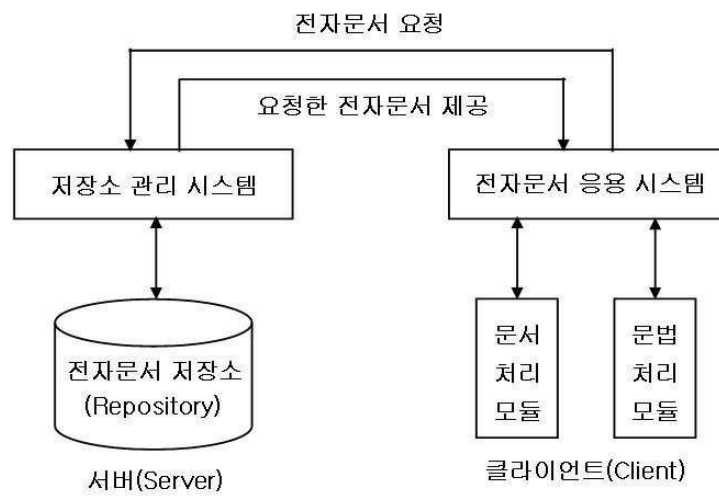
- <93> 도 6a를 참조하면, DeweyID/ Interval 방법은 act[1] 전에 act 노드를 삽입할 때, 6,605개의 노드를 재레이블링 해야 한다. Prime 방법은 재레이블링 해야 하는 노드의 수는 기존 방법보다 줄지만 노드의 삽입시마다 SC 값을 다시 계산해주어야 하기 때문에 계산 시간이 많이 걸린다. Prime의 재계산에 필요한 시간은 DeweyID/Interval 보다 1/3정도 재계산 시간이 필요하다. 본 발명은 재레이블링되어야 하는 노드의 개수가 거의 없음을 알 수 있다.
- <94> 도 6a 및 도 6b의 그래프에서 보여주듯이, 본 발명은 순서에 민감한 노드의 삽입 시, 기존의 노드를 재레이블링 할 필요가 없고 기존의 어느 값도 재계산할 필요가 없기 때문에 갱신 비용이 매우 절감되며 재 레이블링 처리 시간도 현저히 감소하는 것을 알 수 있다.
- <95> 도 7은 본 발명에 의한 레이블 사이즈를 다른 기법과 비교한 결과 및 삭제된 레이블을 재사용하는 경우의 결과를 함께 제시한 그래프이다.
- <96> 도 7을 참조하면, QED 기법에 의한 레이블과 본 발명(IBSL)에 의한 레이블 사이즈는 도 7a에 제시된 바와 같이 삭제에 대한 기법이 있는 경우(Reuse)와 없는 경우에 현격히 차이가 나는 것을 확인할 수 있다.
- <97> 본 방법발명은 또한 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다.
- <98> 컴퓨터가 읽을 수 있는 기록매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피디스크, 광데이터 저장장치 등이 있으며, 또한 캐리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현되는 것도 포함한다. 또한 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어 분산방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.
- <99> 이제까지 본 발명에 대하여 그 바람직한 실시예를 중심으로 살펴보았다. 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다.
- <100> 그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 본 발명의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그와 균등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

도면의 간단한 설명

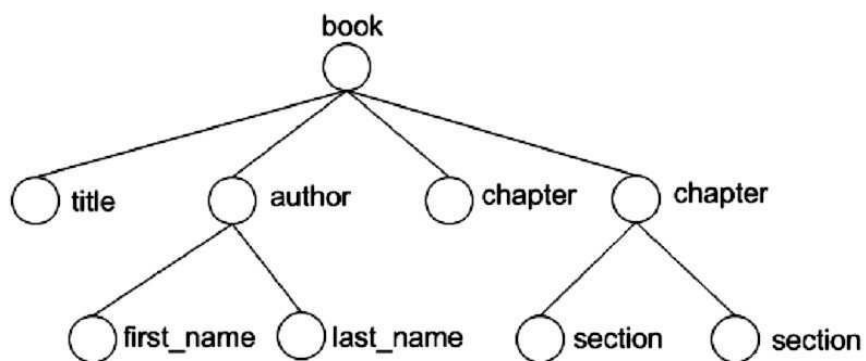
- <101> 도 1a는 XML 문서를 이용하는 클라이언트 서버 시스템을 도시한 것이다.
- <102> 도 1b는 순서가 있는 XML 데이터의 구조를 트리 형태로 나타낸 것이다.
- <103> 도 2는 본 발명의 일실시 예에 따른 XML 문서의 노드에 대한 레이블링 장치의 구성을 제시한 도면이다.
- <104> 도 3은 본 발명의 일실시 예에 따른 XML 문서의 노드에 대한 레이블링 방법의 흐름을 제시한 도면이다.
- <105> 도 4는 도 2 내지 도 3의 레이블링 기법을 이용하여 XML 문서의 노드를 레이블링한 예시도이다.
- <106> 도 5는 본 발명을 통해 얻어지는 질의 처리 시간을 나타내는 그래프이다.
- <107> 도 6a 및 도 6b는 각각 본 발명을 통해 얻어지는 재레이블링 되는 노드 수와재레이블링 처리 시간을 나타내는 그래프이다.
- <108> 도 7은 본 발명에 의한 레이블 사이즈를 나타내는 그래프이다.

도면

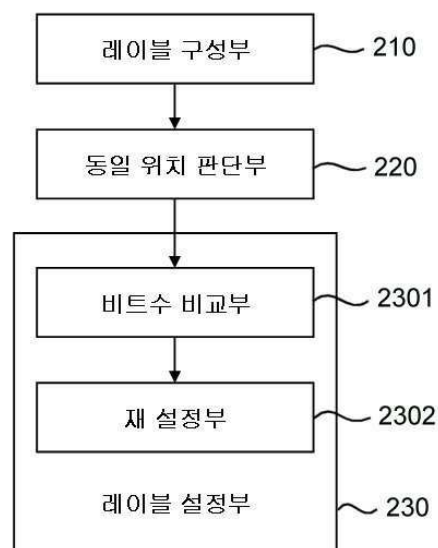
도면1a



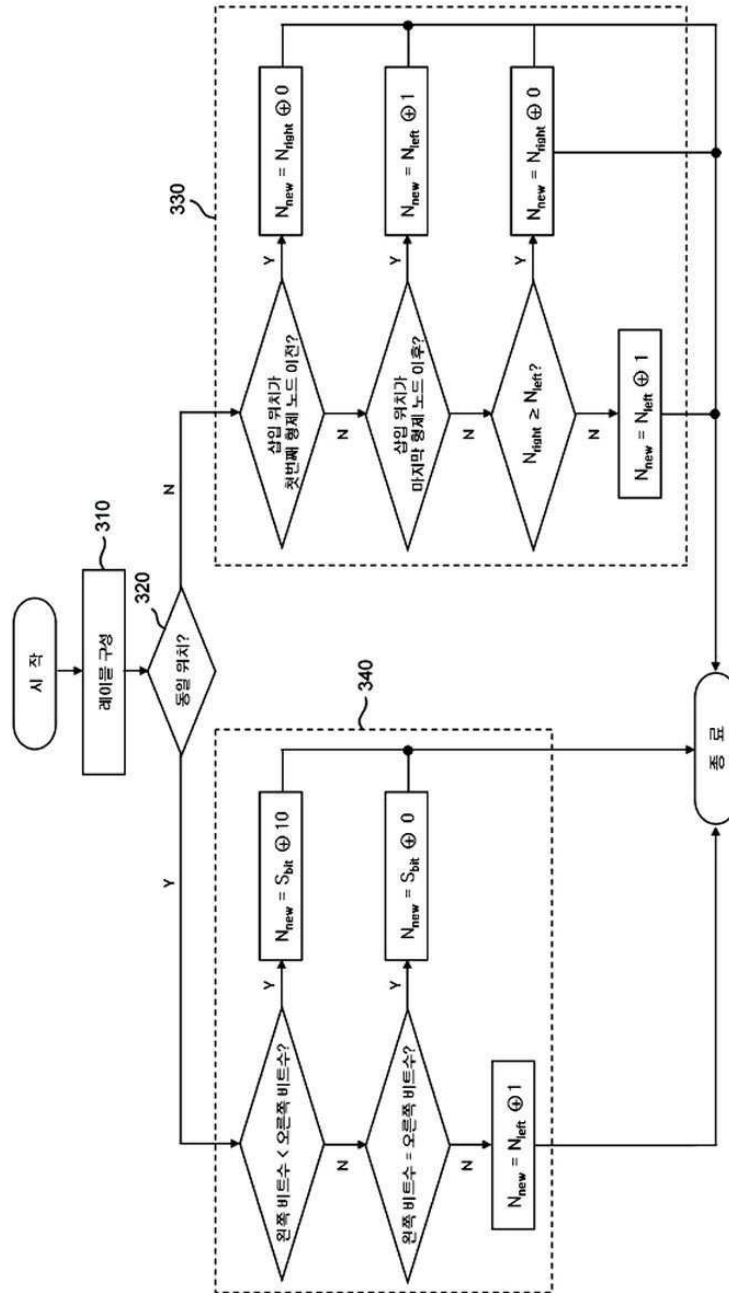
도면1b



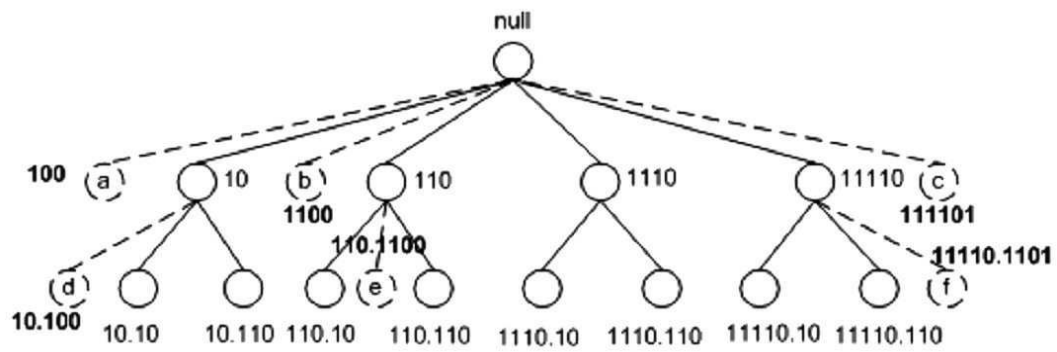
도면2



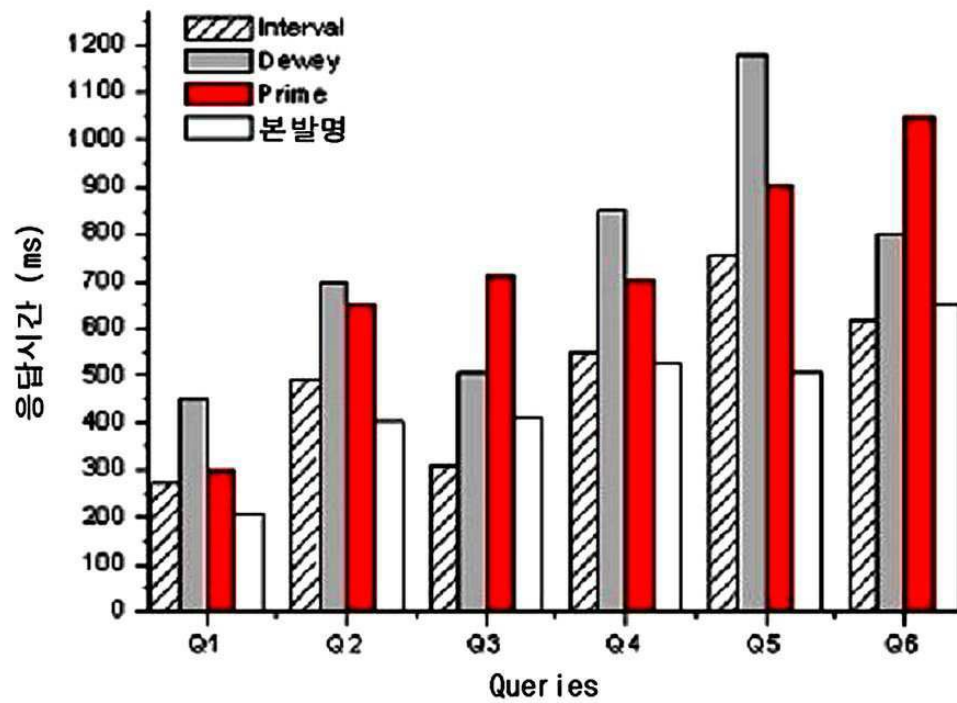
도면3



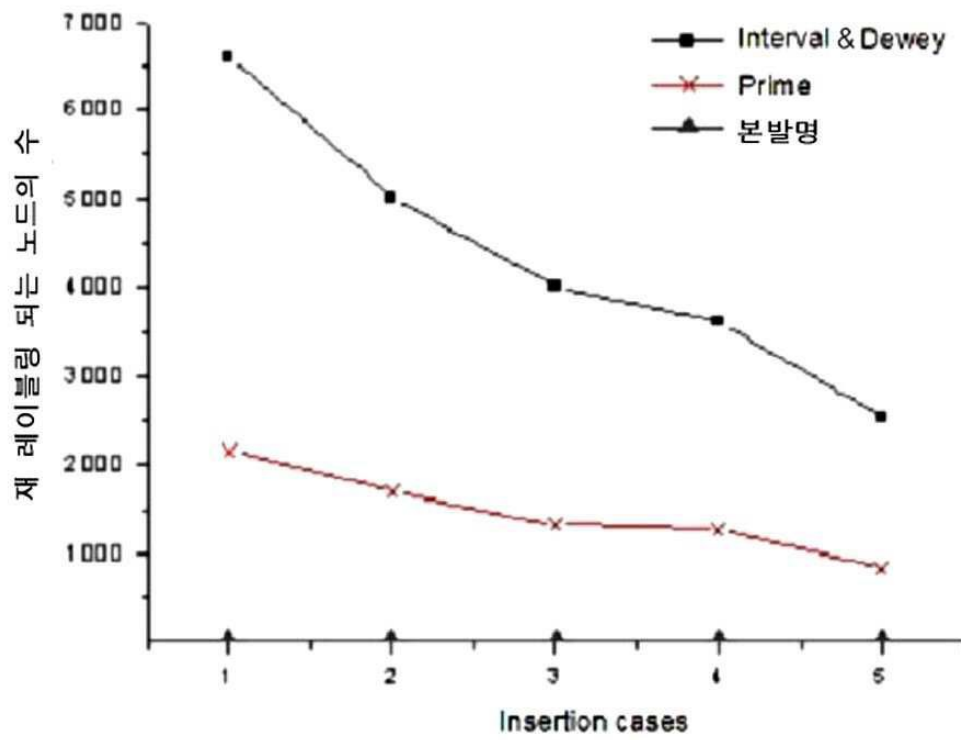
도면4



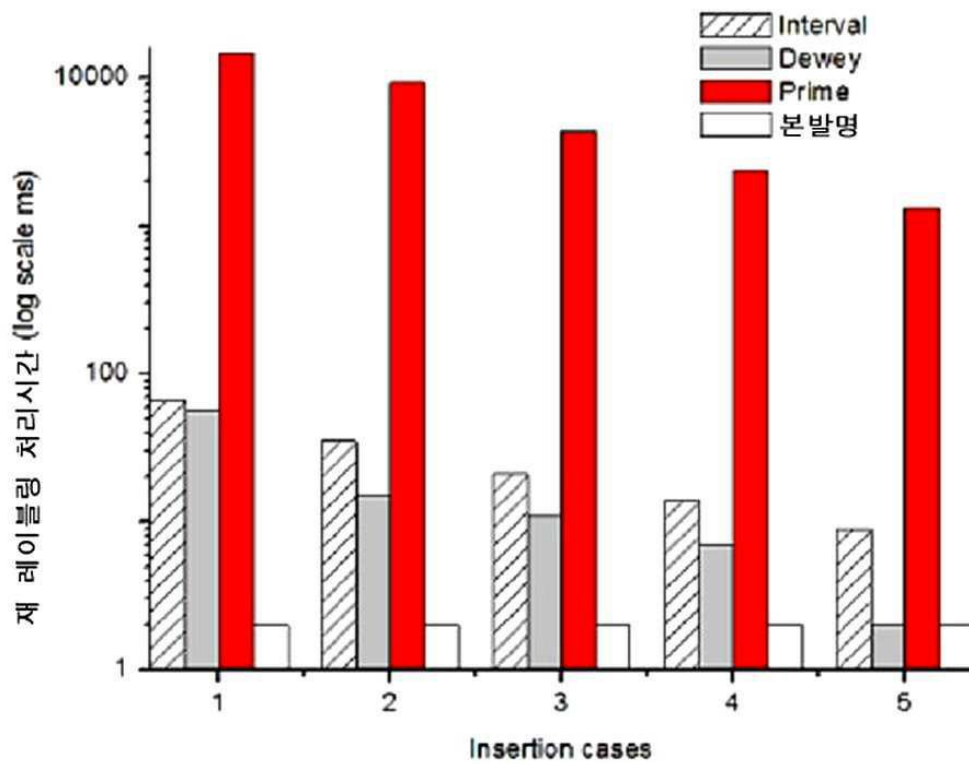
도면5



도면6a



도면6b



도면7

