NCPI-MDS:수정된 DTD 간소화 절차를 통한 새로운 Constraints-Preserving Inlining 기법

안성철*, 김영웅*
*한성대학교 컴퓨터공학과
e-mail : {ascsg21, yukim}@hansung.ac.kr

NCPI-MDS:New Constraints-Preserving Inlining Method with Modified DTD Simplification

Sung-Chul Ahn*, Young-Ung Kim*
* Dept. of Computer Engineering, Han-Sung University

요 약

XML(eXtensible Markup Language)은 웹 상의 데이터를 표현하고, 교환하기 위한 표준 언어로써 XML 로 표현된 문서를 관계형 데이터베이스 관리시스템(RDBMS)에 저장하고 관리하는 기법에 대한 연구[1][2][3]가 활발히 진행되어 왔다. 이러한 연구들은 DTD(Document Type Definition)를 입력 받아 해당 DTD 에서 관계형 스키마를 추론하는 기법을 사용한다. 하지만, 기존의 연구들은 DTD 간소화 절차를 적용하기 때문에 DTD 내에서 추론될 수 있는 의미적인 부분들이 스키마 생성 시에 보존이 되지 못한다. 또한, 기존의 연구들은 XML 데이터의 내용(content)와 구조(structure) 정보만을 저장하는데 초점이 맞춰져 있기 때문에, XML 문서 저장 시 데이터의 무결성을 보장하기 위해 저장프로시져나 트리거를 이용해야 하는 번거로움이 생긴다.

본 논문에서는 [3]의 연구에서 제시한 Inlining 기법을 기반으로 기존의 Inlining 기법의 문제점인 DTD 에서 추론할 수 있는 의미적인 부분의 손실을 관계형 스키마로 보존하는 방법과 효율적인 릴레이션 생성을 위해 개선된 Inlining 기법을 제시한다.

1. 서론

인터넷 상에서 XML 로 표현된 데이터들이 많아짐에 따라 이를 효율적으로 저장하고 관리하는 기법들에 대한 연구가 활발히 진행되었다. 이러한 연구들은 주로 관계형 데이터베이스 관리시스템을 이용하여 XML 을 저장하는 기법[1][3], 파일 시스템이나 객체지향 DBMS 로 저장하는 기법[4]이 많이 연구되어 왔으며, 대표적인 기법으로는 연구 [1], [3]에서 제시된 Inlining 기법이 존재한다.

Inlining 기법에서는 Input 으로 DTD 를 받아서 관계형 스키마를 추론해 내는 알고리즘을 사용한다. 이 때, DTD 로부터 관계형 스키마에 보존되어야 하는 정보로는 문서의 내용(XML 문서의 실제 데이터), 문서의 구조(XML 문서의 엘리먼트들의 부모-자식 관계 정보), 문서의 의미(XML 문서의 제약조건 정보)가 있다[5]. 그러나, 기존의 Inlining 방법에서는 XML 데이터의 내용이나 구조를 저장하는 기법에 초점이 맞추어져 있기 때문에 DTD 에서 추론해 낼 수 있는 의미적인(semantic) 부분들을 관계형 스키마에 보존하지 못한다. 또한, DTD 의 복잡성(complexity)를 다루기 위해 기존 Inlining 기법들은 DTD 간소화 절차를 거치게 되는데, 이 과정에서 DTD 의 지나친 일반화로 인해 추론해 낼 수 있는 의미적인 부분들이 손실되는 결과가 초래된다. 반면에 연구 [2], [5]에서 제시한 저장 기법은 XML 데이터의 의미적인 부분을 관계형 스키마로 보존할 수 있는 방법을 보여주지만, 명시적으로 DTD 간소화 절차를 보여주지 못하기 때문에 DTD 의 복잡성 문제를 다루지 못한다.

기존의 제시된 Hybrid Inlining 기법들의 문제점으로는 다음과 같은 요소들이 있다.

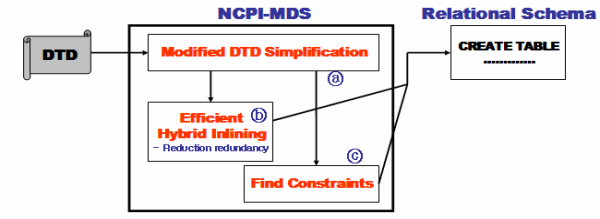
① DTD 간소화 절차를 거치면서 DTD에서 추론할 수 있는 NULL, NOT NULL, Cardinality Constraints 와 같은 DTD의 의미적인 부분들의 손실이 발생

할 수 있다.

② 연구 [1], [3]에서는 DTD의 의미적인 부분을 관계형 스키마로 mapping 시킬 수 있는 방법을 제시하지 않고 있다.

③ 연구 [2]에서는 DTD로부터 추론될 수 있는 의미적인 부분을 보존을 하고 있지만, 효율적인 릴레이션 생성에 관한 고려가 없기 때문에 연구 [3]에서 제시한 방법보다 더 많은 릴레이션이 생성이 되고, 데이터의 중복도 많아진다.

본 논문에서는 위의 문제점을 해결하기 위해 [그림 1]과 같은 접근 방법을 보이는 NCPI-MDS(New Constraints-Preserving Inlining with Modified DTD Simplification)기법을 제시한다. NCPI-MDS 기법에서는 기존의 Inlining 기법과 동일하게 Input 으로 DTD를 받고, DTD 에서 추론할 수 있는 의미적인 부분을 보존하기 위해 수정된 DTD 간소화 절차를 거치게 되며(ⓐ), 의미적인 부분들이 보존된 DTD 를 통해 관계형 스키마를 추론하기 위해 기존의 Hybrid Inlining 에서 데이터의 중복을 줄이고, 효율적 릴레이션 생성을 하기 위해 수정된 EHI(Efficient Hybrid Inlining) 알고리즘을 적용하고(ⓑ), DTD 내의 의미적인 부분들을 추론해 냄으로써 이를 결과 관계형 스키마에 보존하게 되는 과정을 거친다(ⓒ).



[그림 1] NCPI-MDS 방법

본 논문의 구성은 다음과 같다. 2 장에서는 Hybrid Inlining 기법에 대해 살펴보고, 3 장에서는 Hybrid Inlining 기법의 문제점을 개선한 NCPI-MDS 기법을 제안하고, 4 장에서는 두 가지 기법에 의해 생성된 릴레이션에 대한 비교분석을 보여준다. 마지막으로 5 장에서 본 논문의 결론을 맺는다.

2. Hybrid Inlining 기법

가. Hybrid Inlining 알고리즘

기존의 Hybrid Inlining 방법의 세부적인 알고리즘은 다음과 같다.

① DTD 문서를 파싱해서 얻어진 DTD 그래프를 기반으로 DFS 탐색을 실행한다.

② 탐색 동안 처음 방문한 노드는 " VISITED" 로 표시하고, 그 노드의 모든 자식 노드들을 방문했다면 언마크한다.

③ 만약 마킹되지 않은 노드가 탐색 동안에 방문되었다면 같은 이름으로 엘리먼트 그래프에 새로운 노드로 추가하고, 방문 전 노드로부터 방문 중인 노드까지의 간선(edge)을 생성한다.

④ 만약, 이미 마킹이 된 노드를 다시 방문하게 된다면, 이전 노드와 방문 중인 노드 사이에 recursive 가 존재하게 되므로 방문 중인 노드에서 이전 노드로의 역방향의 간선을 추가한다.

⑤ 위의 과정을 반복한 후 만들어지는 엘리먼트 그래프를 기반으로 스키마를 생성한다.

Hybrid Inlining 방법에 따라 생성된 엘리먼트 그래프를 기반으로 릴레이션을 생성하는 규칙은 다음과 같다.

① 엘리먼트 그래프의 루트 엘리먼트에 대해서 릴레이션을 생성한다.
- 루트 엘리먼트의 모든 자식 노드들은 릴레이션으로 inlining 된다.
-" *" 나 recursive 가 존재하지 않는 경우에 대해서는 in-degree 가 1 이상이어도 inlining 시킨다.

② in-degree 가 1 인 노드는 다른 노드의 속성으로 inlining 된다.

③ " *" 아래에 있는 노드는 새로운 릴레이션으로 생성한다.

④ recursive 가 존재한다면 관련된 두 노드 중에 하나의 노드를 릴레이션으로 생성한다.

기존의 Hybrid Inlining 기법은 DTD 에서 추론할 수 있는 많은 의미적인 부분들을 보존하지 못하고, 중복 데이터의 발생, 비효율적인 릴레이션 생성과 관련한 문제를 가진다.

3. NCPI-MDS 기법

본 장에서는 DTD 의 의미 정보를 관계형 스키마로 보존하기 위해서 DTD 간소화 절차의 개선점을 제시하고, 기존의 Hybrid Inlining 기법에서의 문제점이었던 데이터의 중복을 제거하면서 효율적인 릴레이션을 생성하기 위한 기법을 제시한다. 또한, DTD 내에서 추론할 수 있는 의미적인 Constraints 를 언급하고, 이를 관계형 스키마로 보존할 수 있는 기법을 제시한다.

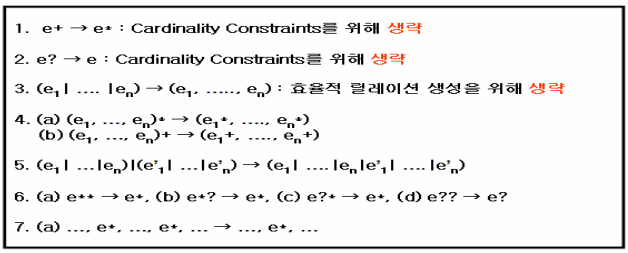
가. DTD 간소화

[그림 3]은 연구 [3]에서 제시되고 있는 DTD 간소화 절차이다.

1. $e^+ \rightarrow e^*$.
2. $e? \rightarrow e$.
3. $(e_1 \mid \cdots \mid e_n) \rightarrow (e_1, \cdots, e_n)$.
4. (a) $(e_1, \cdots, e_n)^* \rightarrow (e_1^*, \cdots, e_n^*)$.
   (b) $e^{**} \rightarrow e*$.
5. (a) $\cdots, e, \cdots, e, \cdots \rightarrow \cdots, e^*, \cdots, \cdots$.
   (b) $\cdots, e, \cdots, e^*, \cdots \rightarrow \cdots, e^*, \cdots, \cdots$.
   (c) $\cdots, e^*, \cdots, e, \cdots \rightarrow \cdots, e^*, \cdots, \cdots$.
   (d) $\cdots, e^*, \cdots, e^*, \cdots \rightarrow \cdots, e^*, \cdots, \cdots$.

[그림 2] DTD 간소화 절차

DTD 에서 추론할 수 있는 정보임에도 불구하고 [그림 2]의 간소화 규칙을 거치면서 정보의 손실을 초래하는 규칙은 규칙 1, 2, 3 이고, 각각 " +" ," ?" ," |" 연산자의 의미를 손실한다. 이러한 의미를 스키마 생성 시에 보존하기 위해 수정된 DTD 간소화 절차는 [그림 3]과 같다.



[그림 3] 수정된 DTD 간소화 규칙

[그림 3]의 수정된 DTD 간소화 절차에서 남겨진 규칙들 또는 추가된 규칙들은 자명한 규칙들만이 남게 된다. 예를 들어, 규칙 7 에서 볼 수 있듯이 {0, ...}의 발생이 가능한 엘리먼트에 {0, ...}의 발생이 가능한 엘리먼트를 더하면 당연히 해당 엘리먼트는 {0, ...}의 발생이 가능하다는 하나의 엘리먼트 정의로 간소화될 수 있으며, 이는 원래의 의미적인 정보인 {0, ...}의 발생이 가능하다는 것이 손실되지 않는다.

나. NCPI-MDS 알고리즘

기존의 Hybrid Inlining 알고리즘을 기반으로 하면서 효율적인 릴레이션 생성을 위해 수정된 NCPI-MDS 알고리즘은 다음과 같다.

① 수정된 DTD 간소화 절차를 거친 DTD 를 Input 으로 내부적으로 DTD 그래프를 생성한다.

② DTD 그래프에서 새로운 릴레이션으로 생성해야 하는 top nodes[2]를 시작으로 DFS 탐색을 한다.

③ top node 를 시작으로 DFS 탐색을 하는 동안 다른 top node 를 만나지 않는 한 모든 도달 가능한 leaf nodes 들을 top node 로 inline 시킨다.

④ DFS 탐색 중 " |" 연산자를 사용하는 노드를 만나면, 이 연산자로 연결된 엘리먼트가 말단 노드일 경우는 모든 엘리먼트를 하나의 칼럼에 저장하기 위해 pcdata 칼럼을 추가하고, 실제 저장된 엘리먼트를 구별하기 위해 nodetype 칼럼을 추가한다. 또한, 말단 노드가 아닌 중간 노드이고 새로운 릴레이션으로 만들어져야 한다면 모든 엘리먼트를 저장하기 위한 통합된 하나의 릴레이션을 생성한다.

⑤ ③ ~ ④까지의 절차를 반복해서 실행한 후 추론된 릴레이션 중 T_1(ID), T_2(ID)의 형태를 갖는 릴레이션이 두 개 이상 존재한다면, 이들을 table(ID, nodetype)의 형태의 하나의 릴레이션으로 통합시킨다.

⑥ 또한, T_1(ID, $PCDATA_1$), T_2(ID, $PCDATA_2$)의 형태를 갖는 릴레이션이 두 개 이상 존재한다면, 이들 릴레이션을 table(ID, nodetype, PCDATA)의 형태의 하나의 릴레이션으로 통합시킨다.

⑦ 마지막으로 " *" 연산자에 의해 만들어지는 릴레이션과 부모 릴레이션 사이의 관계정보를 저장하는 asterisk(parentID, childID, parentType, childType)의 릴레이션을 추가한다. 이 릴레이션을 이용해서 DTD 내에서의 " *" 연산자로 연결된 부모, 자식 노드의 관계정보를 저장한다.

이러한 스키마 생성은 XML 데이터 저장 시에 중복을 제거해 줄 수 있을 뿐 아니라, 일반적인 기본키와 외래키의 조인에 의한 부모 엘리먼트에서 자식 엘리먼트를 탐색하는 하향식 탐색에 있어서 탐색비용을 단축할 뿐만 아니라 자식 엘리먼트에서 부모 엘리먼트를 탐색하는 상향식 탐색에 있어서도 탐색비용을 줄이는데 도움이 된다[3].

다. DTD 내의 의미적인 제약조건[2]

DTD 에서 추론할 수 있는 의미적인 제약조건과 관계형 스키마에서 매핑될 수 있는 제약조건들은 다음과 같다.

도메인 제약조건 : DTD 에서 추론할 수 있는 도메인 제약조건은 속성의 값이 특정 집합의 값으로써 한정될 때 확인할 수 있다. 예를 들어, <!ATTLIST author gender (male | female) #REQUIRED married (yes | no) #IMPLIED>와 같은 엘리먼트 정의에서 author 엘리먼트는 속성으로 gender 와 married 를 가지고, gender 라는 속성은 값으로써 " male" 또는 " female" 중 한 값을 갖고, married 라는 속성은 값으로써 " yes" 또는 " no" 중 한 값을 가질 수 있다. 이러한 특성은 관계형 스키마에서 CHECK 절을 사용하여 보존될 수 있다. 또한, #REQUIRED 와 #IMPLIED 와 같은 속성은 SQL 의 NULL, NOT NULL 로 보존이 될 수 있다.

대응수 제약조건 : DTD 내에서 추론이 될 수 있는 대응수는 총 네 가지로써, [표 1]와 같다.

<표 1> 가능한 Cardinality Relationship

타입	대응수	연산자	제약조건
A	1대{0, 1}	?	NULL, UNIQUE
B	1대{1}		NOT NULL, UNIQUE
C	1대{0, ...}	*	NULL
D	1대{1, ...}	+	NOT NULL

Inclusion Dependencies (IDs) : IDs 는 XML 문서 내의 특정 속성의 값이 다른 속성의 값으로 나타나야만 한다라는 의미로써[2]. 참조무결성의 일반화된 개념이라 할 수 있다. DTD 에서 IDs 를 추론할 수 있는 요소는 ID, IDREF, IDREFS 속성이 있다. 이는 관계형 스키마의 기본키와 외래키의 개념으로 보존이 될 수 있다.

4. 비교분석

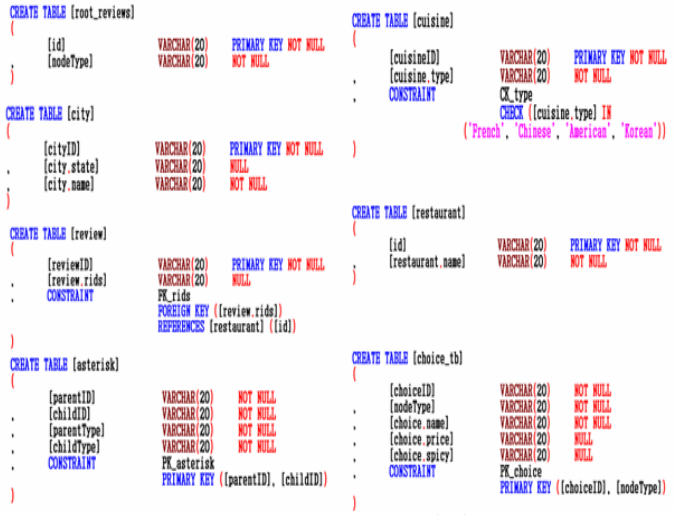
본 장에서는 [그림 4]의 DTD 에 대해 NCPI-MDS 기법을 사용해서 생성되는 릴레이션과 기존의 Inlining 기법으로 생성되는 릴레이션을 비교 분석한다.

```
<!ELEMENT root          (city*)>
<!ELEMENT city          (state?, restaurants, reviews*, name)>
<!ELEMENT state         (#PCDATA)>
<!ELEMENT reviews       (review*)>
<!ELEMENT review        (#PCDATA)>
<!ATTLIST review        rids    IDREF   #IMPLIED>
<!ELEMENT restaurants   (cuisine*)>
<!ELEMENT cuisine       (restaurant*)>
<!ATTLIST cuisine type  (French | Chinese | American | Korean)
                                #REQUIRED>
<!ELEMENT restaurant    ((appetizer | salad | desert | entree)*, name)>
<!ATTLIST restaurant    id      ID      #REQUIRED>
<!ELEMENT appetizer     (name, price)>
<!ELEMENT salad         (name, price)>
<!ELEMENT desert        (name, price)>
<!ELEMENT entree        (name)>
<!ATTLIST entree        spicy   CDATA   #IMPLIED>
```

(그림 4) DTD 문서

[그림 4]의 DTD 를 따르는 XML 문서를 저장하기 위해 NCPI-MDS 기법과 기존 Hybrid Inlining 기법의 방법대로 DTD 를 기반으로 관계형 스키마를 생성하면 [그림 5], [그림 6]과 같다.

(그림 5) Hybrid Inlining 기법



(그림 6) NCPI-MDS 기법

[그림 5]의 Inlining 기법은 [그림 6]의 기법보다 더 많은 릴레이션을 생성할 뿐만 아니라, DTD 에서 추론할 수 있는 의미적인 부분을 릴레이션에 보존하지 못한다. 예를 들어, [그림 6]에서는 DTD 에서 추론할 수 있는 SQL CHECK 제약조건을 릴레이션에 보존한다. 또한, IDREF 와 같은 속성 정의의 SQL 의 외래키 제약조건으로 보존하고, #REQUIRED, #IMPLIED 와 같은 정의는 NULL 제약조건으로 보존된다. DTD 에서 추론할 수 있는 제약조건들을 보존하는 것에 더해서 본 논문에서 제시하는 NCPI-MDS 기법은 효율적인 릴레이션 생성을 위해 root 엘리먼트와 reviews 엘리먼트를 위해 릴레이션을 하나의 릴레이션으로 통합하고, "|" 연산자에 의해 정의된 엘리먼트도 또한 각각의 엘리먼트를 하나의 릴레이션으로 통합한다. 또한, "*" 연산자로 정의된 부모 엘리먼트와 자식 엘리먼트의 관계 정보를 저장하기 위해 asterisk 릴레이션을 도입함으로써, 데이터의 중복 저장을 방지하고 탐색에 도움을 준다.

5. 결론

XML 문서를 RDBMS 에 저장하기 위한 기법 중 Hybrid Inlining 기법은 DTD 로부터 보존이 될 수 있는 많은 의미적인 정보들을 손실할 뿐만 아니라, 실제 XML 문서를 저장할 때 많은 데이터의 중복이 발생하는 문제점을 보인다.

이에 본 논문에서는 DTD 로부터 추론될 수 있는 의미적인 정보들을 보존하기 위해 수정된 DTD 간소화 절차를 제시하고, 이를 관계형 릴레이션으로 매핑할 수 있는 방법을 제시하였으며, 데이터의 중복을 방지하고, 데이터 질의 시 탐색에 도움이 될 수 있는 효율적 릴레이션 생성을 다룬 NCPI-MDS 기법을 제시하였다.

향후 연구 방향은 본 논문에서 제시한 NCPI-MDS 기법에 의해 생성된 릴레이션에 대해 질의응답시간이나 조인수와 같은 다양한 측정기준을 가지고 성능평가가 이루어 져야 할 것이다. 또한, 릴레이션을 추론하기 위해 DTD 를 사용하는 것이 아니라 DTD 의 문제점을 극복하기 위해 제시된 XML Schema 를 사용하여 릴레이션을 추론하는 기법도 연구되어야 한다.

참고문헌

[1] J. Shanmugasundaram et al. " Relational databases for querying XML documents: Limitations and opportunities" . In Proc. Of VLDB, Edinburgh, Scotland, 1999.

[2] D. Lee, and W. W. Chu, " Constraints-Preserving Transformation from XML Document Type Definition to Relational Schema" , International Conference on Conceptual Modeling / the Entity Relationship Approach, 2000.

[3] S. Lu, Y. Sun, M. Atay, and F. Fotouhi, " A New Inlining Algorithm for Mapping XML DTDs to Relational Schemas" , Proc. of the the 1st International Workshop on XML Schema and Data Management, 2003.

[4] S. A. T. Lahiri, J. Widom, " Ozone : Integrating Structured and Semistructured Data" , In proceedings of DBPL Conf., 1999.

[5] Y. Chen, S. Davidson, and Y. Zheng, " Constraint Preserving XML Storage in Relations" , In WebDB, 2002.