2000년 한국정보처리학회 추계 학술발표논문집 제7권 제2호

XML 데이터의 효율적인 DTD 추출

양은주, 박경현, 류근호
충북대학교 컴퓨터과학과
{ejyang, khpark, khryu}@dblab.chungbuk.ac.kr

An Efficient Technique for Extracting DTD from XML Data

Eun Joo Yang, Kyung Hyun Park, Keun Ho Ryu
Dept. of Computer Science, Chungbuk National University

요 약

기존의 데이터를 인터넷상에서 XML 데이터 형태로 전송 시 부하를 줄이기 위해 DTD가 없는 형태로 전송하지만, 전송 받은 XML 데이터에 대한 저장 및 질의처리를 최적화하기 위해서는 DTD 추출이 필요하다. 따라서 이 논문에서는 반구조적 데이터의 특징을 갖는 XML 데이터에 대한 DTD를 추출하기 위해 기존의 데이터로그(DataLog)를 이용하여 반구조적 데이터의 최소 경계 스키마를 추출하는 방법보다 향상된 방법인 시뮬레이션을 이용한 최소 경계 스키마 추출 방법을 제시함으로써 보다 효율적인 DTD 추출을 가능하게 하는 방법을 제시한다.

1. 서론

인터넷상에서 정보를 효과적으로 관리, 교환, 검색하기 위해 개발된 XML은 표준화된 텍스트 형식으로 의학, 경영, 법률, 판매 자동화, 디지털 도서관, 전자상거래 등 다양한 분야에 활발히 응용되고 있다. 이러한 XML 데이터는 자기 서술적(self-describing)이고 고정된 스키마가 존재하지 않는다(schemaless)는 점에서 반구조적 데이터(semistructured data)의 성격을 지니고 있다[1]. 반구조적 데이터란 기존의 데이터베이스에서 요구하는 일관되고 정형화된 구조를 갖지 않으며, 미리 정의된 스키마 구조를 가지고 있지 않는 데이터를 말한다. 이런 관점에서 XML은 반구조적 데이터의 한 인스턴스가 될 수 있다[2].

또한 반구조적 데이터는 주어진 데이터 인스턴스에 대해 하나 이상의 스키마가 존재할 수 있는데 이러한 구조 정보 추출 비용은 시스템에 많은 부담을 주게 된다. 이러한 문제를 해결하기 위해 지금까지 다양한 스키마 추출 기법들이 제안[3, 4, 5, 6]되어 왔으며, 그 중 대표적인 것으로서 데이터가이드(DataGuide)와 데이터로그(Datalog)가 있다.

이 논문에서는 반구조적 데이터의 특징을 지니고 있는 XML 데이터로부터 DTD를 추출하기 위해 반구조적 데이터의 스키마 추출기법인 데이터로그를 이용하는 기존의 방법보다 향상된 방법인 시뮬레이션을 이용한 최소 경계 스키마 추출 방법을 제시함으로써 보다 효율적인 DTD 추출 방법을 제시한다.

2. 관련연구

반구조적 데이터의 스키마 추출기법은 반구조적 데이터에 대한 저장 및 질의를 최적화 함으로써 사용자의 편의성을 도모하기 위해 많은 연구가 이루어져 왔다.

반구조적 데이터의 최소 경계 스키마를 추출하는 데이터로그는 프로그램의 최대 고정점을 이용하여 타입을 추출하는데 이와 같이 단순히 최대 고정점을 이용하여 타입을 추출하게 되면 많은 수의 타입이 생성되며 경우에 따라서는 실제 데이터와 비슷한 양만큼의 타입이 생성되는 경우도 있기 때문에 이것을 해결하는 방법으로 클러스터링(clustering)방법을 이용하여 타입을 줄이고 있다[6].

반구조적 데이터에 대한 최대 경계 스키마를 생성하는 방법인 데이터가이드는 반구적 데이터의 스키마를 간결(concise)하고 정확하게(accurate) 요약한다. 이는 데이터베이스의 구조를 볼 수 있고 질의를 생성하고, 통계 정보와 샘플 값을 저장할 수 있는 동적 스키마이다[3].

Lore 프로젝트[7]는 반구조적 데이터에 대한 대표적인 연구 중 하나로 이 논문에서는 Lore 프로젝트에서 소개된 반구조적 데이터 모델인 OEM(Object Exchange Model)과 데이터가이드를 이용하여 최대 경계 스키마를 추출하고 있다.

XTRACTOR[8]에서는 DTD를 반구조적 데이터의 스키마와는 다른 것으로 간주하고 XML 문서로부터 DTD를 추출하기 위해 반구조적 데이터의 스키마 추출 기법을 이용하지 않고 있다. 이는 [3, 6, 9]에서 추출한 스키마가 간선들의 집합이나 순서적인 나열로 표현될 뿐 임의의 정규 표현식에 해당되는 DTD를 추론하는데는 사용될 수 없다는 사실에 기인한다. 즉, 반구조적 데이터에서 추출한 스키마는 임의의 정규식으로 표현될 수 없고 이는 정규식으로 표현되는 DTD를 표현하기엔 불가능하다고 언급하고 있다. 따라서 XTRACTOR는 기존의 반구조적 데이터의 스키마 추출 기법을 적용하지 않고 [10]과 [11]에서 제안한 MDL(Minimum Description Length) 기법을 이용하여 DTD를 추론한다. 그러나 데이터 중심의 XML문서(data centric XML documents)인 경우, 문서 중심의 XML문서(document centric XML documents)와는 달리 반구조적 데이터와 밀접한 관련이 있기 때문에 반구조적 데이터의 최대/최소 경계 스키마 추출 기법을 이용하면 XTRACTOR에서 소개하는 방법보다 효율적으로 데이터 중심의 XML 문서로부터의 DTD 추출이 가능해진다.

3. 스키마 추출

3.1 최대 경계 스키마 추출

데이터가이드는 데이터베이스 구조를 간결하고, 정확하게 나

타내는 스키마로 정의된다. 주어진 데이터베이스는 루트 객체에 의해서 구별된다. 데이터가이드는 데이터 소스의 모든 유일한 레이블 경로를 데이터 소스에 나타나는 빈도에 상관없이 한번만 기술한다. 데이터 소스에 나타나지 않는 경로는 데이터가이드에 나타나지 않는다. 데이터가이드에 대한 이러한 특성은 반구조적 데이터의 최대 경계 스키마 추출을 가능하게 해준다.

```
<school>
 <student>
  <name>
   <first> Donggil </first>
   <last> Hong </last>
  </name>
  <class>
   <department> Economics </department>
   <grade> junior </grade>
   <major> macroeconomics </major>
  </class>
  <email> hong@econo.aaa.ac.kr </email>
  <email> hong@bbb.net </email>
 </student>
 <student>
  <name>
   <first> Gildong</first>
   <last> Hong </last>
  </name>
  <class>
   <department> Computer Science </department>
   <grade> junior </grade>
   <major> Database </major>
  </class>
  <phone> 02-700-0001 </phone>
  <phone> 016-400-0001 </phone>
 <email> gdhong@cs.aaa.ac.kr </email>
 </student>
 <student>
  <name>
   <first> Eunjin</first>
   <last> Kim </last>
  </name>
  <class>
   <department> Computer Engineering </department>
   <grade> junior </grade>
   <major> Database </major>
  </class>
  <phone> 02-777-0001 </phone>
  <email> ejkim@ce.aaa.ac.kr </email>
 </student>
</school>
```

그림 1. 데이터 중심의 XML 문서

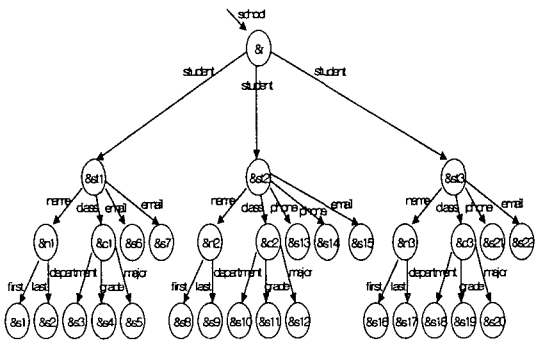

그림 2. 반구조적 데이터 모델

그림 1은 중첩된 태그 엘리먼트들로 구성되어 있는 XML 문서를, 그림 2는 그림 1에 대응되는 반구조적 데이터 모델을 나타낸 것이다. 이에 대한 최대 경계 스키마 추출은 다음과 같다.

그림 2의 데이터 그래프에서의 root노드 {&r}과 &dg1로 초기화되어 있는 데이터가이드에서 시작하여 &r의 모든 자식노드에 대해서 <label, oid> 순서쌍을 원소로 하는 집합 P를 생성하고 P 집합의 원소들에 대해서 label이 같은 oid들의 집합들 즉, <label,{oid, oid…}>가 원소가 되는 집합 T를 생성한다. 집합 T의 모든 원소들은 새롭게 데이터가이드에 삽입될 노드(예:&dg1, &dg2, &dg3 …)를 생성하고 이것을 데이터가이드에 삽입한다. 이때 삽입 노드를 생성하기 전에 T의 원소인 <label,{oid, oid, …}>순서쌍에서 {oid, oid, …}가 해시 테이블에 존재하지 않는다면 해시 테이블에 삽입하고 노드를 생성하여 현재 노드에 연결한다. 그러나 이미 해시 테이블에 존재한다면 새로운 노드를 생성하지 않고 해시 테이블에 존재하는 노드 값에 해당하는 데이터가이드의 노드를 현재 노드에 연결한다. 이에 대한 알고리즘은 그림 3과 같다.

```
targetHash = global empty hash table
dg = global oid

MakeDataGuide(o) {
  dg = NewObject();
  targethash.Insert({o}, dg);
  RecursiveMake({o}, dg);
}

RecursiveMake(t1, d1) {
  p = set of <label, oid> children pairs of each object in t1;
  foreach (unique label l in p) {
    t2 = set of oids paired with l in p
    d2 = targetHash.Lookup(t2);
    if (d2 != nil) {
      add an edge from d1 to d2 with label l
    } else {
      d2 = NewObject();
      targetHash.Insert(t2, d2);
      add an edge from d1 to d2 with label l
     RecursiveMake(t2, d2)
    }
}
```

그림 3. 데이터가이드 알고리즘

예를 들어 ({&r}, &dg1)순서쌍은 데이터가이드를 생성하기 위한 초기 상태이다. 여기서 &r에 대한 P값을 생성하면 P = {<student, &st1>, <student, &st2>, <student, &st3>}이 생성이 되고 P에서부터 집합 T를 생성하면 T={<student, {&st1, &st2, &st3}이 된다. 현재 T는 1개의 원소를 가지는데 이 원소에 대해서 해시 테이블에 {&st1, &st2, &st3}에 해당하는 값이 존재하지 않기 때문에 새로운 노드 &dg2를 생성하고 해시 테이블에 이를 삽입하고 현재 노드 &dg1에 student라는 레이블 명을 갖는 간선으로 연결한다. 위와 같은 방법을 순환(recursion)을 이용하여 현재 상태가 단말 노드의 집합으로 이루어질 때까지 반복한다. 그림 4는 그림 2를 바탕으로 이러한 과정을 거쳐 추출한 데이터가이드를 보여준다.

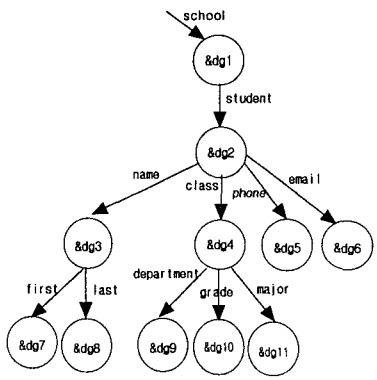

그림 4. 데이터가이드를 이용한 최대 경계 스키마 그래프

3.2 최소 경계 스키마 추출

3.2.1 시뮬레이션(simulation)

주어진 데이터 그래프를 D라 하고 그 데이터 그래프에 대한 스키마 그래프를 S라 할 때 그래프 시뮬레이션에 대한 정의는 다음과 같다. 만약, D에서 S로의 시뮬레이션이 존재한다면 $D \preceq S$로 표시한다. 예를 들어 D의 노드에서 S로의 노드로 가는 이진 릴레이션 $\preceq$은 다음의 (1), (2)를 만족한다. (1) root(D)$\preceq$root(S), (2) $u \preceq u'$ 이고 $u \xrightarrow{a} v$에서 a가 간선 레이블이라면 $u' \xrightarrow{p} v'$가 존재해야 되며 $v \preceq v'$가 되어야 한다. 이때 u는 주어진 임의의 정점 v에 대한 자식 정점들의 원소이다.

이를 자세히 설명하면, 집합 D, S에 대한 이진 릴레이션 $\preceq$은 카티션 프로덕트(cartesian product) $D \times S$의 부분집합이다. $\preceq$가 이러한 릴레이션이라면 $(x, y) \in \preceq$에 대해 $x \preceq y$로 표기할 수 있다. 주어진 방향 레이블 그래프 (V, E)에 대해서 각각의 간선 레이블 l은 V에 대한 이진 릴레이션 [l]을 유도한다. 이처럼 x에서 y로 향하는 l-레이블 간선이 있을 때 x[l]y로 표현한다. 이 조건은 아래 그림 5에 잘 나타나있다. 그림5에서 R은 이진 릴레이션 $\preceq$이다. 시뮬레이션이 되기 위한 $\preceq$ 조건은 실선으로 주어진 패턴이 주어지면 점선으로 되어 있으면서 그래프를 완성하는 y2을 발견하게 된다. 데이터 그래프의 간선 x1[l]y1이 간선 x2[l']y2에 의해서 시뮬레이트 된다고 한다. 릴레이션 $\preceq$이 시뮬레이션이고 이것의 역 $\preceq^{-1}$ 또한 시뮬레이션이 되면 이중시뮬레이션(bisimulation)이라 한다.

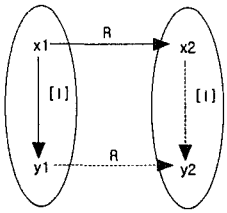

그림 5. 시뮬레이션 다이어그램

3.2.2 시뮬레이션을 이용한 최소 경계 스키마 추출

그래프 시뮬레이션은 두 개의 입력 그래프에 대한 일치성 검사에 유용하게 이용되는 방법이다. 이 방법으로 데이터 그래프 G가 주어지면 두 개의 똑같은 G에 대한 그래프 시뮬레이션을 이용하여 타입정보를 추출할 수가 있다. 즉, 하나의 그래프를 대상으로 그래프가 내포하고 있는 타입 정보를 시뮬레이션을 이용하여 추출할 수가 있다. 타입 추출을 위한 알고리즘 기술을 위해서 필요한 몇 가지 정의를 하면 주어진 그래프 G의 임의의 정점 v에 대한 sim(v)는 v를 시뮬레이션하는 정점들의 집합이다. 즉, v가 가지고 있는 출력 간선을 포함하는 정점들을 의미한다. 주어진 임의의 정점 v에 대해서 부모 정점들과 자식 정점들을 post(v) = {ul (v, u)E}와 pre(v) = {ul(u, v)E}로 각각 정의할 수 있는데 여기서 E는 그래프에 속하는 간선들의 전체 집합을 의미한다. 데이터로그를 이용하는 방법에서 순환조건으로 어떤 객체의 내향 프리디킷의 확장에 대해서 만족하는지를 검사하였다. 이러한 조건이 시뮬레이션을 하는 과정에서도 검사가 되어져야 하기 때문에 임의의 정점v에 대해서 remove(v)라는 것을 정의해야 하는데 이는 어떤 정점v에 대해서 v의 pre(v)에 속하지 않는 객체들의 집합을 의미한다. 만약 어떤 정점v에 대해 초기에 sim(v)의 집합을 얻었을 때, 이 sim(v)에 속하는 모든 객체들은 간선을 통한 객체들간의 관계가 고려되지 않는다. 그러므로 어떤 정점v가 주어지면 그 정점v의 pre(v)의 원소 u에 대한 sim(u)를 구하고 sim(u)에서 remove(v)에 해당하는 객체들을 제거함으로써 간선을 통한 객체돌 상호간의 관계를 고려할 수가 있다. 다음의 알고리즘 그림 6은 이와 같은 이론을 기반으로 그래프 시뮬레이션을 이용하여 스키마를 추출하는 방법을 나타낸다.

```
foreach (node in graph G) {
  labels = getNodeLabels(v);
  foreach(v' in graph G) {
    labels' = getNodeLabels(v');
    if (labels ⊆ labels')
      sim(v).add(v');
  }
  remove(v) = pre(v) - pre(sim(v));
}
prevsim(v) = V;
while (vertex v such that remove(v) ≠ ∅) {
    {assert for v, remove(v) = pre(prevsim(v)) - pre(sim(v));}
    foreach (node v in graph G) {
      u = pre(v);
      foreach (p in remove(u)) {
        foreach (w in remove(v)) {
          if (w ∈ sim(p)) {
            sim(p) = sim(p) - w;
            foreach (w'' in  pre(w)) {
              if (post(w'') ∩ sim(p) = ∅)
                remove(p) = remove(p) ∪ w'';
            } // foreach
          } // if
        } // foreach
      } // foreach
    } // foreach
    prevsim(v) = sim(v);
} // while
```

그림 6. 시뮬레이션 알고리즘

이러한 시뮬레이션 알고리즘을 통해 그림 2에 대한 최소 경계 스키마를 추출한 결과는 그림 7과 같다.

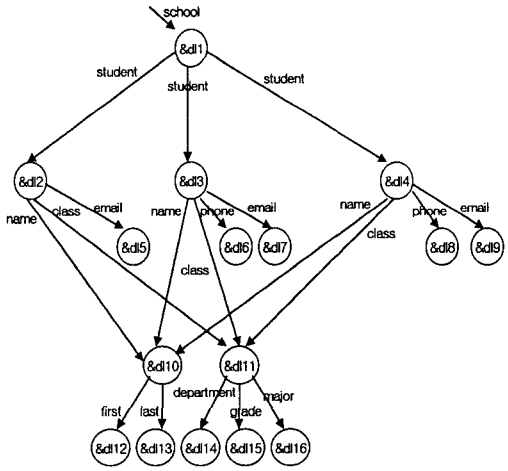

그림 7. 시뮬레이션을 이용한 최소 경계 스키마 그래프

시뮬레이션 알고리즘의 초기화는 데이터로그규칙으로부터 만들어진 타입 릴레이션과 같다. 또한 스키마 추출의 결과를 가지고 스키마 그래프를 생성하기 위해서는 각각의 sim(v)집합에 대응하는 규칙들이 존재해야 된다. 따라서 스키마 추출을 위한 초기화 방법과 스키마 추출 결과에 대한 스키마 그래프 생성은 데이터로그 알고리즘과 동일하다. 그러나 시뮬레이션 알고리즘은 데이터로그의 최대고정점을 대신하여 시간비용을 줄이는 역할을 한다.

4. XML 문서의 DTD 추출

지금까지 반구조적 데이터의 스키마 추출 기법을 이용하여 XML 문서로부터 최대 경계 스키마와 최소 경계 스키마를 추출하였다. 최대 경계 스키마의 경우 주어진 데이터 그래프에 대해서 타입을 구분할 때 모호성이 발생하지 않는 반면 최소 경계 스키마에서는 모호성이 발생하게 된다. 예를 들어 그림 7에서 &dl1을 기준으로 student를 통해 도달할 수 있는 노드는 &dl2, &dl3, &dl4로 간선상의 레이블만을 가지고서는 타입을 구성하는데 있어서 모호성이 발생하게 된다.

따라서 같은 레이블을 가지는 간선들을 통합하여 이러한 모호성을 제거함으로써 최대 경계 스키마와 최소 경계 스키마를 얻게 되면 주어진 데이터 그래프로부터 중복이 되는 레이블에 대한 정보를 얻어내야 한다.

예를 들어 그림 2에서 보면 &st1인 노드는 레이블이 email인 출력 간선이 2개 존재하고 &st2인 노드는 레이블이 phone인 출력 간선이 2개 존재함을 알 수 있다. 이것은 문서상에 해당 엘리먼트가 여러 번 중복되어 표현됨을 의미하기 때문에 해당 엘리먼트는 DTD상에서 반드시 "*" 혹은 "+"으로 표현되어야 한다.

이러한 정보를 얻은 후, 추출한 최대 경계 스키마와 최소 경계 스키마를 비교하여 중복되는 부분과 중복되지 않은 부분으로 구분한다. 여기서 중복되는 부분은 반드시 문서상에 나타나야 하는 부분을 나타내고 중복되지 않는 부분은 문서상에 나타날 수도 나타나지 않을 수도 있음을 암시한다. 따라서 레이블이 phone인 간선은 입력 간선의 레이블이 student인 노드로부터 여러 개의 출력간선으로 표현될 수 있고 또는 존재하지 않을 수도 있기 때문에 DTD문서 내에 phone*로 표시되어야 한다.

또한 레이블이 email인 경우는 반드시 문서 내에 존재해야 하고 여러 번 중복되어 존재 할 수도 있기 때문에 다시 말해서 한 번 이상은 문서에 존재하기 때문에 DTD문서 내에 email+로 표시되어야 한다. 그림 8은 DTD 추출에 필요한 이러한 정보를 포함한 스키마 그래프를 보여준다.

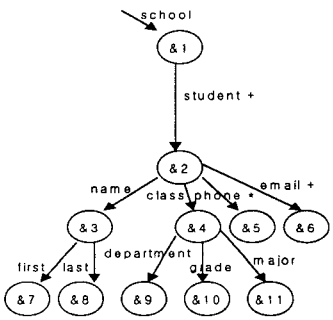

그림 8. 그림 1에 대한 스키마 그래프

일단 DTD를 추출하는데 필요한 정보를 포함한 이러한 그래프를 얻게 되면 깊이 우선 탐색 방법(DFS)을 통하여 DTD를 추출하게 된다. 그림 9는 그림 8의 스키마 그래프로부터 추출한 그림 1에 대한 DTD를 보여준다.

```
<!ELEMENT school (student+)>
<!ELEMENT student (name, class, phone*, email+)>
<!ELEMENT name (first, last)>
<!ELEMENT first (#PCDATA)>
<!ELEMENT last (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
<!ELEMENT class (department, grade, major)>
<!ELEMENT department (#PCDATA)>
<!ELEMENT grade (#PCDATA)>
<!ELEMENT major (#PCDATA)>
<!ELEMENT email (#PCDATA)>
```

그림 9. 그림 1에 대한 DTD

5. 결론 및 향후연구

이 논문에서는 XML 데이터에 대한 저장 및 질의를 최적화함으로써 사용자의 편의를 도모하는데 도움을 주는 DTD를 추출하고자 XML 데이터가 반구조적 데이터의 특징을 갖는다는 점을 기반으로 반구조적 데이터 스키마 추출 기법을 이용하여 XML 문서로부터 DTD를 추출하는 기법을 살펴보았다. 반구조적 데이터는 주어진 데이터 인스턴스에 대해 하나 이상의 스키마가 존재할 수 있는데 이들 스키마는 최대 경계 스키마와 최소 경계 스키마로 구분할 수 있고 데이터가이드를 이용하여 최대 경계 스키마를 추출할 수 있다. 최소 경계 스키마를 추출하는 기존의 방법은 데이터로그를 이용하는 것이다. 그러나 이 논문에서는 최소 경계 스키마를 추출하기 위해 데이터로그를 이용하는 방법보다 효율적인 방법인 시뮬레이션을 이용한 최소 경계 스키마 추출 방법을 제시함으로써 보다 효율적인 DTD 추출 방법을 제시했다.

이 논문에서의 DTD 추출 대상은 데이터 중심의 XML 문서(data centric XML documents)이기 때문에 문서 중심의 XML 문서(document centric XML documents)에서 중요시되는 속성(attribute)과 하위 엘리먼트(subelement)의 구분, 엘리먼트간의 순서, 그리고 엘리먼트들 간의 링크 등은 이 논문에서 고려하지 않았다. 따라서 향후 연구로는 데이터 중심의 XML 문서뿐만 아니라 문서 중심의 XML 문서에 적합한 DTD추출에 대한 연구가 필요하다.

참고문헌

[1] S. Abiteboul, P. Bunneman, D. Suciu, Data on the Web: From Relations to Semistructured Data and XML, Morgan Kaufmann, 1999.

[2] S. Abiteboul, Querying semi-structured data, In Proc. of the Intl. Conf. On Database Theory (ICDT), 1997.

[3] R. Goldman, J. Widom, DataGuide : Enabling Query Formulation and Optimization In Semistructured Databases, In Proc. of the 23rd VLDB Conference Athens, Greece, 1997.

[4] P. Buneman, S. Davidson, G. Hillebrand, and D. Suciu, A query language and optimization techniques for unstructured data, In SIGMOD, pages 505-516, Montreal, 1996.

[5] D. Calvanese, G. Giacomo, and M. Lenzerini, What can Knowledge representation do for semi-structured data?, In Proc. of the 15th National Conf. on Artificial Intelligence(AAAI-98).

[6] S. Nestorov, S. Abiteboul, R. Motwani: Extracting Schema from Semistructured Data, In SIGMOD, pages 295-306, 1998.

[7] J. McHugh, S. Abiteboul, R. Goldman, D. Quass, and J Widom, Lore: A Database Management System for Semistructured Data, SIGMOD Record, 26(3), September, 1997.

[8] M. Garofalakis, A. Gionis, R. Rastogi, S. Seshadri, K. Shim, XTRACT: A System for Extracting Document Type Descriptors from XML Documents, In Proc. of the ACM SIGMOD international Conf. On Mangement of Data, Dallas, Texas, 2000.

[9] M. Fernandez and D. Suciu, Optimizing regular path expressions using graph schemas, In Proc. of the Intl. Conf. on Database Theory(ICDT), 1997.

[10] A. Brazma, Efficient identification of regular expressions from representative examples, In Proc. of the Ann. Conf. on Computational Learing Theroy(COLT), 1993.

[11] P. Kilpelainen, H. Mannila, and E. Ukkonen, MDL learning of unions of simple pattern languages from positive exmamples, In Proc. of the European Conf. on Computational Learning Theroy (EuroColt), 1995.