Lecture 14: Database Theory in XML Processing

Thursday, February 15, 2001

Outline

- Skolem Functions
- XML Publishing

Skolem Functions

In Logic

- Vocabulary: $R_1, \ldots, R_k, g_1, \ldots, g_p$
  - Recall that Mathematical Logic talks about relations $R_1, \ldots, R_k$ *and* functions $g_1, \ldots, g_p$
- The problem: given a formula φ, decide whether it is *satisfiable:*
  - φ true in some model $\mathbf{D} = (D, R_1, \ldots, R_k, g_1, \ldots, g_p)$

Skolem Functions

- Write φ in ***prenex normal form***:

$$\varphi = \exists y_1 \forall x_1 \forall x_2 \exists y_2 \forall x_3 \exists y_3 \exists y_4 . \theta(x_1, x_2, x_3; y_1, y_2, y_3, y_4)$$

- Replace existential quantifiers with Skolem functions (next)

Skolem Functions

$$\varphi = \exists y_1 \forall x_1 \forall x_2 \exists y_2 \forall x_3 \exists y_3 \exists y_4 . \theta(x_1, x_2, x_3; y_1, y_2, y_3, y_4)$$

- Becomes:

$$\forall x_1 \forall x_2 \forall x_3 . \theta(x_1, x_2, x_3; f_1(), f_2(x_1, x_2), f_3(x_1, x_2, x_3), f_4(x_1, x_2, x_3, x_4))$$

- Then delete universal quantifiers:

$$\varphi' = \theta(x_1, x_2, x_3; f_1(), f_2(x_1, x_2), f_3(x_1, x_2, x_3), f_4(x_1, x_2, x_3, x_4))$$

Skolem Functions

In Logic

Theorem φ is satisfiable iff φ' is satisfiable.

- φ true in some model:
  - $\mathbf{D} = (D, R_1, \ldots, R_k, g_1, \ldots, g_p)$
- iff φ' true in some model:
  - $\mathbf{D'} = (D, R_1, \ldots, R_k, g_1, \ldots, g_p, f_1, f_2, f_3, f_4)$

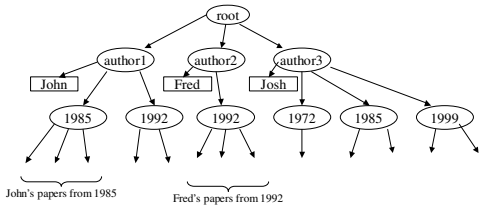
Skolem Functions in Databases

Author(aid, name, email), Paper(pid, title, year), AP(aid, pid)

- Want to construct Webpages declaratively
  - WebPage(wid) - all webpage id's
  - Text(wid, value) - some text associated to web pages

7

Skolem Functions in Databases



A great Website, with papers grouped by year !

8

Skolem Functions in Databases

Author(aid, name, email), Paper(pid, title, year), AP(aid, pid)

WebPage(Root()) :-
WebPage(Author(aid)) :- Author(aid, _, _)
Text(Author(aid), name)) :- Author(aid, name, _)
WebPage(Year(aid,year)) :- Author(aid, _, _), AP(aid, pid), Paper(pid, _, year)
WebPage(Paper(aid, pid, year)) :- ……

- Author(aid) "means": create a new object, for each value of aid
- Year(aid,year) "means": create a new object, for each value of aid and year

9

Skolem Functions in Databases

- A closer look:
  Text(Y, name)) :- Author(aid, name, _)
- Unsafe, because of Y

$$\forall aid.\forall name.\forall z.\forall Y.(Text(Y, name) \Leftarrow Author(aid, name, z))$$

10

Skolem Functions in Databases

- But let us change the rules of the game:
  - "all variables in the head that don't occur in the body are existentially quantified (not universally)"

$$\forall aid.\forall name.\forall z.\exists Y.(Text(Y, name) \Leftarrow Author(aid, name, z))$$

- Becomes equivalent to a Skolem function:
  Text(f(aid, name, z), name) :- Author(aid, name, z)

11

Skolem Functions in Databases

- f's arguments depend on the order in which we write the quantifiers

$$\forall name.\exists Y.\forall aid.\forall z.(Text(Y, name) \Leftarrow Author(aid, name, z))$$

- Becomes:
  Text(f(name), name) :- Author(aid, name, z)
- Idea in databases: write the Skolem functions and their arguments explicitly:
  Text(author(aid), name) :- Author(aid, name, z)
- Makes possible *object fusion*, when we reuse the Skolem function

12

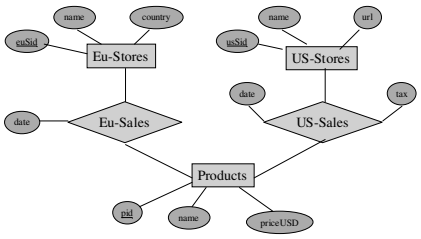
Publishing XML Data

- mediator for exporting legacy data to XML
- define XML view declaratively
  - virtual XML view
  - materialized XML view

13

SilkRoute: an Example

Legacy data in E/R:



14

SilkRoute: an Example

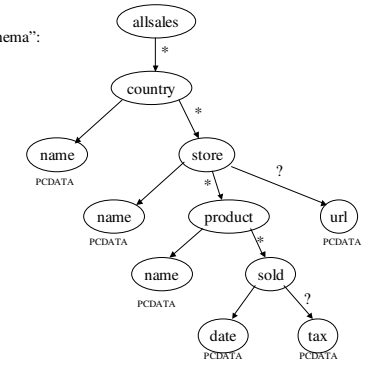
- XML view

```
<allsales>
  <country>  <name> France </name>
             <store> <name>  Nicolas </name>
                     <product> <name> Blanc de Blanc </name>
                               <sold> 10/10/2000 </sold>
                               <sold> 12/10/2000 </sold>
                               …
                     </product>
                     <product>…</product>…
             </store>….
  </country> …
</allsales>
```

- In summary: group by country → store → product

15

Output "schema":



16

SilkRoute Query

```
{ FROM    EuStores $S, EuSales $L, Products $P
  WHERE  $S.euSid = $L.euSid  AND $L.pid = $P.pid
  CONSTRUCT
    <allsales()>
      <country ID=c($S.country)>
        <name> $S.country </name>
        <store ID=s($S.euSid)>          /* means: s($S.country, $S.euSid) */
               <name> $S.name </name>
               <product ID=p($P.pid)>   /* same: add arguments above */
                    <name> $P.name </name>
                    <price>  $P.priceUSD </price>
               </product>
        </store>
      </country>
    <allsales>
}                                /* union….. */
```

```
  …. /* union */
{ FROM    USStores $S, EuSales $L, Products $P
  WHERE  $S.usSid = $L.euSid  AND $L.pid = $P.pid
  CONSTRUCT
    <allsales()>
      <country ID= c("USA")>            /* object fusion here */
        <name> USA </name>
        <store ID= s($S.euSid)>             /* object fusion here */
               <name> $S.name </name>
               <url> $S.url </url>
               <product ID= p($P.pid)>        /* object fusion here */
                    <name> $P.name </name>
                    <price>  $P.priceUSD </price>
                    <tax> $L.tax </tax>
               </product>
        </store>
      </country>
    <allsales>
}
```

Notes on the Syntax

- All Skolem functions inherit the arguments of their parent.
  - Why ?
- Have explicit Skolem functions:

```
CONSTRUCT … <store ID=s($S.euSid)>
CONSTRUCT … <store ID=s($S.euSid)>   /* fuse ! */
CONSTRUCT … <store ID=t($S.euSid)>   /* don't fuse ! */
```

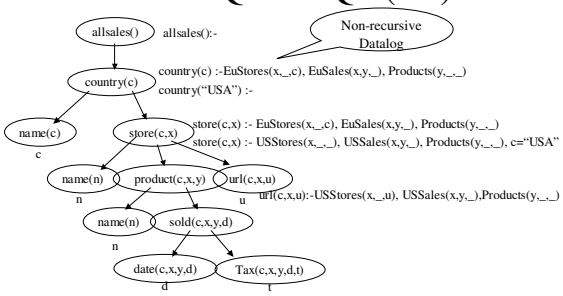
19

Users Ask XML-QL Queries

- find names, urls of all stores who sold on 1/1/2000

```
WHERE <allsales/country/store>
            <product/sold/date> 1/1/2000 </>
            <name> $X </>
            <url> $Y </>
      </>
CONSTRUCT <result> <name> $X </>
                        <url> $Y </>
                  </result>
```

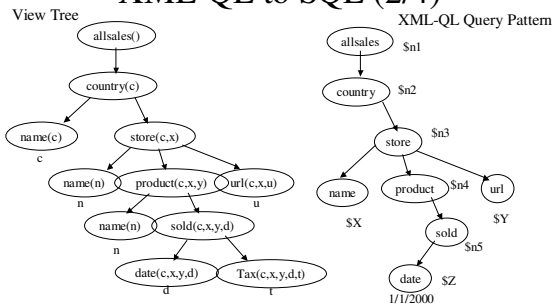
XML-QL to SQL (1/4)



Step1: construct the **View Tree**

21

XML-QL to SQL (2/4)



Step2: "evaluate" the XML-QL pattern(s) on the view tree

22

XML-QL to SQL (3/4)

- Step 3: for each answer:

$n1	$n2	$n3	$n4	$n5	$X	$Y	$Z
Allsales()	Country(c)	Store(c,x)	Product(c,x,y)	Sold(c,x,y,d)	n	u	d

  - Collect all datalog rules
  - Rename variables properly
  - Do query minimization on the result
  - Obtain…

23

XML-QL to SQL (4/4)

```
( SELECT S.name, S.url
  FROM USStores S, USSales L, Products P
  WHERE S.usSid=L.usSid AND L.pid=P.pid AND  L.date='1/1/2000')

     UNION

( SELECT S2.name, S2.url
   FROM EUStores S1, EUSales L1, Products P1
         USStores S2, USSales L2, Products P2,
 WHERE S1.usSid=L1.usSid AND L1.pid=P1.pid AND  L1.date='1/1/2000'
    AND S2.usSid=L2.usSid AND L2.pid=P1.pid
    AND S1.country="USA" AND S1.euSid = S2.usSid)
```

24