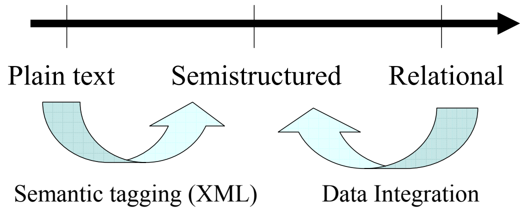
Formal languages problems in semistructured data management

S. Afonin
Moscow State University

Semistructured data

- ***Semistructured data*** (SSD) are the data with irregular, rapidly changing or even unknown structure.



SSD Formal models

- Data structure: tree, ordered tree, directed graph
- Integrity constraints: data schemes (e.g. XML DTD), additional constraints (path equivalencies)
- Query language: functional, regular path queries (CRPQ, XPath)

Graph representation

The directed edge-labelled graph $D=\langle V, \Sigma, E\rangle$ where

V is the set of vertices, and

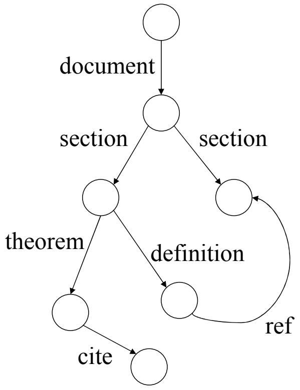
$E\subseteq\langle V\times \Sigma \times V\rangle$ is the set of Σ-labelled edges

is called a *semistructured database.*

Vertices correspond to domain-specific objects, edges represent relations between these objects.

Edge labels reflect relation type.

Graph representaion



\begin{document}

\section{A}\label{secA}

\section{B}

\begin{definition}

see also \ref{secA}

\end{definition}

\begin{theorem}[\cite{Ivanov}]

\end{theorem}

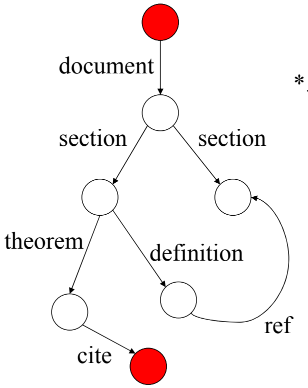
\end{document}

Regular path queries

- A *query* is a regural expression (language)
- Query *result* is the set of all pairs of the graph vertices such that there exists at least one path between them labeled by a word from a given language
- Example: **X** *.subsection.(theorem+definition).cite **Y**
- Conjunctive regular path queries:
  - **X** *. theorem **Y, Y** cite **Z**

FDPW-2005

SSD Examples



Query:

*.section.(theorem+definition).cite

*.(sub)*section.(theorem+definition).cite

The whole graph may need to be searched during query evaluation, which is inefficient

FDPW-2005

View-based query processing

- Let us assume that we know the results for the queries E1,...,Ek.
- Can this data be used during evaluation of an arbitrary query?
- Is it possible to compute the result of a query using the result of view queries only?

FDPW-2005

Definitions

- An ***alphabet*** is a finite non-empty set of em symbols
- A ***word*** is a finite sequence of symbols; e - is the ***empty word***
- A ***language*** is an arbitrary set of words

Language operations

- Union
  - L1+L2 = { u | u in L1 or u in L2 }
- Concatenation
  - L1.L2 = { uv | u in L1 and v in L2 }
- Iteratioln
  - L* = {e} + L + L.L + L.L.L + ...
- Regular langluges is the minimal class of languages which contains *singletons* and closed under union, concatenation and iteration.

Language representation problem

- For finite set {E1,...,Ek} of regular languages, and a subset T of language operations (**+**,**.**,*****) is it decidable whether or not a given language R may be constructed from {Ei} using a finite number of operations from T?
  - T={concatenation} - language factorization
    R = a*+a*ba*b(a+b)*   E1=a*+a*b(a+b)*b   E2=(bab)*
    R=E1.E2.E2.E1.E1
  - T={**+**,**.**,*****} - maximal rewriting
- Language representation problem is decidable (K.Hashiguchi, 1982)

Representation & query evaluation

- R = a*+a*ba*b(a+b)*
  E1=a*+a*b(a+b)*b   E2=(bab)*

  R=E1.E2.E2.E1.E1

- E1
  (1, 2)
  (1, 4)
  (2, 3)
  (3, 1)
- E2
  (1, 3)
  (2, 1)
  (3, 2)
  (2, 2)

E1 E2 E2 E1 E1

1 --> 2 --> 1 --> 3 --> 1 --> {1,4}

...

...

Semigroups of regular languages

- Regular languages are closed under concatenation
- (S, **.**)=<E1,...,Ek> - finitely generated semigroup
- A query R may be represented in terms of {Ei} if and only if the language R belongs to (S, **.**)
- The membership problem is decidable (K.Hashiguchi)

Query rewriting under constraints

- **<u>Theorem</u>** The set of all possible representations of an element R is a regular language.
  - E=(e+b*a)(b+ab*ab*a)*   (S,**.**)=<E>   R=E.E.E

    R=(E.E.E).E*
- **<u>Open Problem</u>** For given set Q1,...Qn of "the most popular queries" find the <u>minimal number</u> of semigroup generators E1,...,Ek, such that all the languages Qi belongs to the semigroup

Language equations

- For given regular languages L (query) and E (view) find a regular language X such that

  L=EX
- If the equation has a solution then it has the unique maximal solution, which is a regular language (L.Kari)
- Maximal solution contains all solutions
- Minimal solution contains no solutions
- If the equation has a solution then it has at least one minimal solution (L.Kari)

Maximal and minimal solutions

- Maximal solution drawback: redundancy
  - a*b = a*.X   X=a*b is the maximal solution

    X=b   is a solution as well
- Infinitely many minimal solutions
  - a* = (e+a) X
    - X = (aa)* and X=e+a(aa)*   are minimal solutions
    - X = {0, 2, 4,..., 2n, 2n+1,   2n+3, 2n+4, 2n+6,.... }

      2n+2

Open Problems

- Is there exists an algorithm for minimal solution finding?
- Is there exist regular languages L and E such that the equation L=EX has no regular minimal solutions?

Thank you