Semi-structured data

Semi-structured data

- Data is not just text, but is not as well-structured as data in databases
- Occurs often in web databanks
- Occurs often in integration of databanks

Semi-structured data - properties

- irregular structure
- implicit structure
- partial structure
- a posteriori 'data guide' versus a priori schema
- large data guides

Semi-structured data - properties

- It should be possible to ignore the data guide upon querying
- Data guide changes fast
- object can change type/class
- difference between data guide and data is blurred

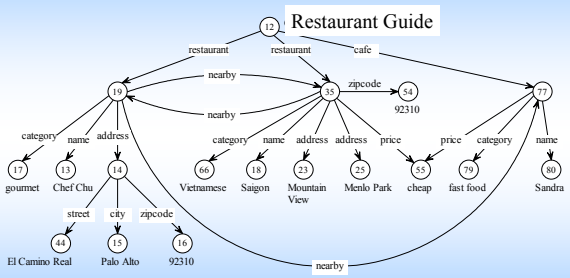
Semi-structured data - model

- network of nodes
- object model (oid)
- query: path search in the network

OEM (Object Exchange Model)

- Graph
- Nodes: objects
  - oid
  - atomic or complex
    - atoms: integer, string, gif, html, …
    - value of a complex object is a set of object references (label, oid)
- Edges have labels
- OEM is used by a number of systems (ex. Lorel)

OEM example



Lorel query language

1. Find all places to eat Vietnamese food

```
select P
from RestaurantGuide.% P
where P.category grep "ietnamese"
```

2. Find the names and streets of all restaurants in Palo Alto

```
select R.name, A.street
from RestaurantGuide.restaurant{R}.address A
where A.city = "Palo Alto"
```


Lorel query language

3. Find all restaurants to eat with zipcode 92310

```
select RestaurantGuide.restaurant
where
RestaurantGuide.restaurant(.address)?.zipcode = 92310
```

Wildcards and variables

? - 0 or 1 path
+ - 1 or more paths
* - 0 or more paths
# - any path
% - 0 or more chars

- object variables
select P from Guide.% P
select A from #.address{A}
- path variables
select Guide.#@P.name

Data Guides

- A structural summary over a databank that is used as a dynamic schema
- Is used in query formulation and optimization
- Is often created a posteriori
- Properties:
  - concise
  - accurate
  - convenient

Data Guides - definitions

- Label path: sequence of labels L1.L2. … .Ln
- Data path: alternating sequence of labels and oid:s L1.o1.L2.o2. … .Ln.on
- Data path *d* is an instance of label path *l* if the sequences of labels are identical in *l* and *d*.

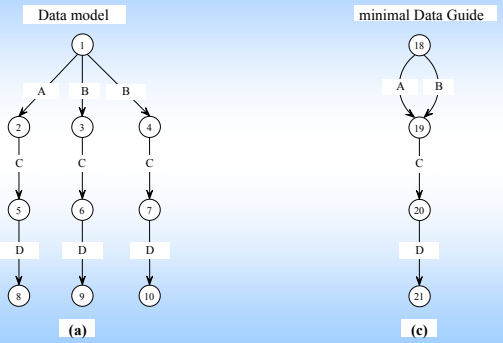
Data Guides - definitions

- A data guide for object *s* is an object *d* such that every label path of *s* has exact one data path instance in *d*, and each label path in *d* is a label path of *s*.

Data Guides

- A databank can have several data guides
- Minimal data guides
  the smallest data guides

Data Guides - example



Minimal Data Guides

- Concise
- May be hard to maintain
  Example: child node for 10 with label E

Strong Data Guides

Intuitively:

”label paths that reach the same set of objects in the data model = label paths that reach the same objects in the data guide”

Strong Data Guides - definitions

An object *o* can be reached from *s* via *l* if there is a data path of *s* that is an instance of *l* and that has *o* as last oid

(L1.o1.L2.o2. … Ln.o)

The target set for label path *l* in object *s* is the set of objects that can be reached from *s* via *l*. *Notation: T(s,l)*

L(s,l): set of label paths of *s* that have the same target set in *s* as *l*.

Strong Data Guides - definitions

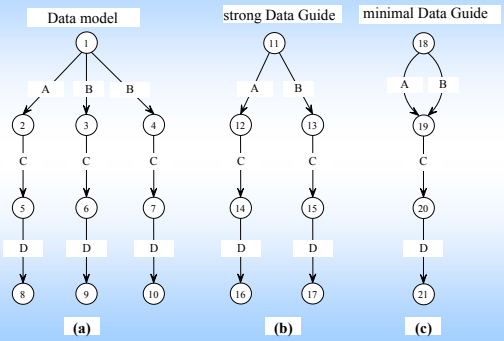
Definition:

d is a strong data guide for *s* if

for all label paths *l* of *s* it holds that *L(s,l) = L(d,l)*

There is a 1-1-mapping between target sets in the data model and nodes in a strong data guide.

Data Guides - example



Strong Data Guides - algorithm

Implementation:

- Traverse data model depth-first.
- Each time you find a new target set for label path *l*, create a new object in the data guide.

If the target set is already represented in the data guide, do not create a new object, but link to the existing object.

Strong Data Guides - use

- Easier to maintain
- Used as path index for query optimization

Semi-structured data - exercises

Exercise 1

- Represent the relations below using the OEM data model.

r_id	name
r1	Hamlet
r2	Normandie
r3	McDonald's

Restaurants

c_id	name
c1	Linkoping
c2	Norkoping

Cities

r_id	c_id	street
r1	c1	Storgatan
r2	c1	St.Larsgatan
r3	c2	Kungsgatan

Restaurants&Cities

Exercise 2

- Using the data model from the previous question, formulate the following queries using Lorel:
  - find all the restaurants that are located in Linkoping
  - find the address (city and street) of the “Hamlet” restaurant
  - list the restaurants by city (equivalent of GROUP BY)

Exercise 3

- Write 4 simple queries in Lorel that illustrate the use of coercion in the following types of comparison:
  - string type against integer type;
  - value against atomic object
  - value against complex object
  - value against set of objects

Explain how coercion works in each case

Exercise 4

- Draw the strong Data Guide for the restaurant guide data model below.