Data Consistency Verification through Model Checking Techniques

Fabio Mercorio (fabio.mercorio@unimib.it)
University of Milano-Bicocca - C.R.I.S.P. Research Centre, Italy

Introduction

Growing relations between citizens and public administrations generate a lot of data

- Data are often *longitudinal*:
  - Repeated observations (events) of the same subject at multiple time points generate a *sequence of* events for the subject;
  - Longitudinal databases allow one to observe and measure how the data change along the time.
- Data quality of enterprise and public administration databases is very low [Batini and Scannapieco, 2005][Redman, 1998];

- Data Quality is described by many dimensions e.g., *accuracy, consistency*;
- *Consistency* describes the violation of semantic rules defined over a set of data items.

EventId	ShipID	City	Date	Event Type
e_1	S01	Venice	12th April 2011	checkin
e_2	S01	Venice	15st April 2011	checkout
e_3	S01	Lisbon	30th April 2011	checkin
e_4	S01	Barcelona	5th May 2011	checkin
e_5	S01	Barcelona	8nd May 2011	checkout
...	...	...	...	...

Table: Travel Plan of a Cruise Ship

Data is used for decision making purposes, e.g. in *Business Intelligence Systems*:
☹ Derive information from low quality database may lead to dangerous or wrong decisions;
☺ **Data Quality Analysis** and **Improvement** techniques are required before using data.

State of the art

Mainly, three categories of techniques deal with data quality problems:

record linkage: Uses alternative (and more trusted) data sources to improve the quality;

error localisation and correction: Typically exploits functional dependencies [Fan, 2008] to detect errors and looks for a *repair*, i.e. another database which is consistent and minimally differs from the original one [Chomicki, 2005];

consistent query answering: Looks for consistent answers from inconsistent data, i.e. the focus is on automatic query modifications and not on fixing the source data. An answer is consistent if appears in every possible repair of the original database [Arenas,2003].

Aim of the research

- Exploit formal methods developing *techniques* for:

Data Quality Analysis: The Robust Data Quality Analysis (RDQA) evaluates and helps to improve the quality level achieved after a cleansing intervention.

Sensitivity Analysis: Quantitatively estimates the impact that cleansing interventions may have on aggregate indicators computed on the data.

What Model Checking can do?

- Given a system model described by:
  - State variables whose evaluation determines a state;
  - Transition relations specifying actions leading the system from a state to another one.
- Given a property to be verified in any state of the system.

An *explicit* Model Checker:

1. explores all the feasible system configurations (reachable states);
2. verifies the properties in each state;
3. returns a sequence of actions describing how the system reaches a state which violates the property (if any).

How to Perform Data Analysis via Model Checking?

Main Idea: Express the problem of Data Consistency verification as a Model Checking Problem.

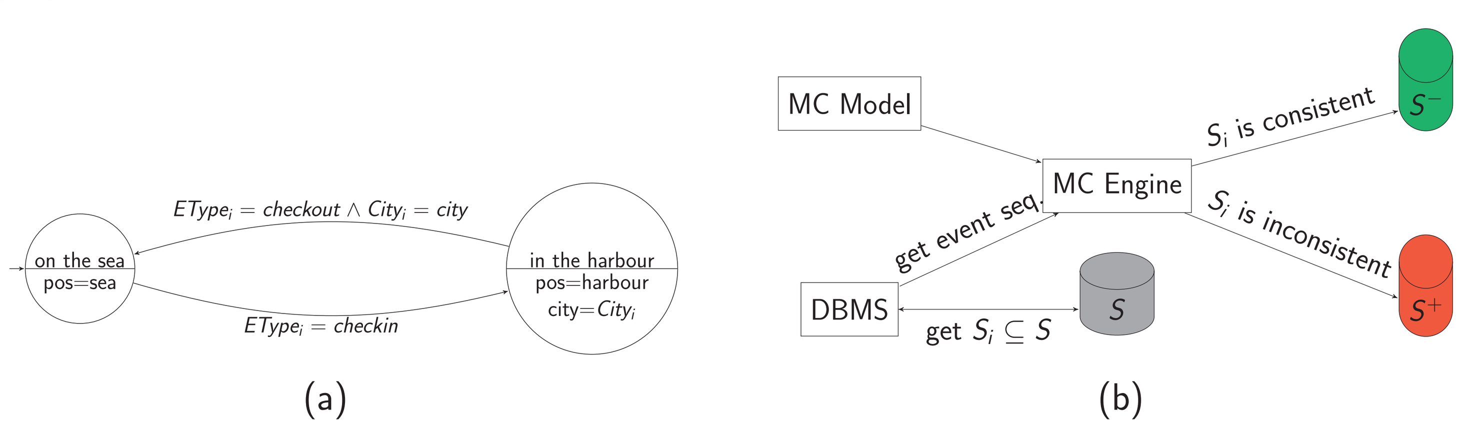

Figure: (a) A Graphical representation of the Travel Plan of a Cruise Ship domain. The lower part of a node describes how the system state evolves when an event happens. An event can be composed by $e_i = (ShipID_i, City_i, Date_i, EType_i)$. (b) A Graphical representation of a model checking based data consistency verification of a database.

1. Define a model representing the *evolution* of the database data S and formalise the *consistency* properties to be verified on it;
2. Exploit model checking to generate a partition S^+, S^- of S as follows:
   - Retrieve each dataset $S_i \subseteq S$ from the database;
   - Solve the Model Checking problem on S_i, i.e. verify the consistency of S_i;
   - If S_i is consistent then insert it into S^-;
   - Otherwise S_i is affected by at least one inconsistency. Return the error-trace and insert S_i into S^+.

The Robust Data Quality Analysis (RDQA)

Given a function *clr* performing consistency verification on a dirty dataset S and generating the cleansed version C, some questions arise:

1. What is the **degree of consistency** achieved through *clr*?
2. Can we **improve the consistency** of the cleansed dataset?
3. Can we be sure that function *clr* does **not introduce any error** in the cleansed dataset?

We use **model checking** to implement a function *ccheck* able to verify the dataset consistency *before* and *after* the *clr* intervention

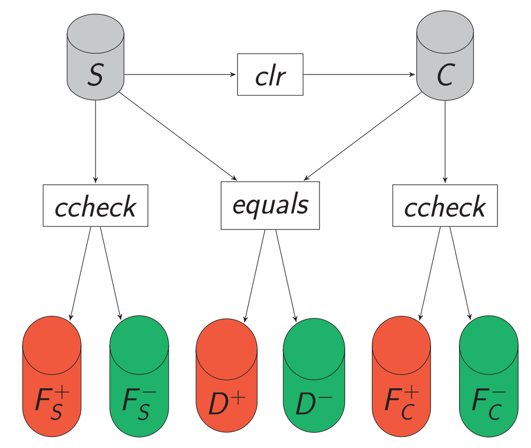

Figure: Schematic view of the RDQA approach.

Conditions			Result
ccheck(S_i)	*equals*(S_i, C_i)	*ccheck*(C_i)	Cardinality
0	0	0	$\|F_S^- \cap D^- \cap F_C^-\|$
0	0	1	$\|F_S^- \cap D^- \cap F_C^+\|$
0	1	0	$\|F_S^- \cap D^+ \cap F_C^-\|$
0	1	1	$\|F_S^- \cap D^+ \cap F_C^+\|$
1	0	0	$\|F_S^+ \cap D^- \cap F_C^-\|$
1	0	1	$\|F_S^+ \cap D^- \cap F_C^+\|$
1	1	0	$\|F_S^+ \cap D^+ \cap F_C^-\|$
1	1	1	$\|F_S^+ \cap D^+ \cap F_C^+\|$

Table: The Double Check Matrix. $ccheck(X) = 0$ means that X is consistent, inconsistent otherwise; $equals(S_i, C_i) = 0$ means S_i is equals to C_i, not equals otherwise.

RDQA works iteratively as follows:

1. Use *clr* to cleanse the source database S generating C;
2. Use *ccheck* to verify the consistency on the source (cleansed) database S (C):
   - F_S+ (F_C+) contains all datasets *violating* the consistency properties;
   - F_S- (F_C-) contains all datasets *satisfying* the consistency properties;
3. Compute differences between S and C:
   - $D+$ contains all datasets *modified* by *clr* function;
   - $D-$ contains all datasets *untouched* by *clr* function;
4. Generate the Double Check Matrix (DCM) and use it to analyse/refine/modify the *clr* behaviour
5. Repeat 1-4 until a satisfying cleansing result is achieved

A Real-World Application: "The Worker Career Archive"

- According to the Italian law, whenever an employer hires or dismisses an employee, a **communication** (or event) is sent to the job archive. The archive content is used to study the labour market, obtaining information about worker career paths and supporting the decision making processes.

A Worker's Event contains:

- The worker ID;
- The event ID and the date;
- The event type: *start*, *cessation*, *extension* or *conversion* of a worker contract;
- The event modality, whether the event is related to a full-time or a part-time contract;
- The contract type wrt the Italian law.

Consistency constraints:

c1: No more than one full-time contract active at the same time;
c2: At most K (i.e, K=2) part-time contracts;
c3: An unlimited term contract cannot be extended;
c4: A contract extension cannot change an existing contract modality and type;
c5: A conversion requires to change a contract modality or type.

- The CRISP research centre aims to evaluate and improve archives data quality.
- The RDQA has been applied to improve the cleansing activities on a worker careers database for an Italian Area, with 1,089,895 mandatory communications regarding 213,566 workers collected in 10 years.

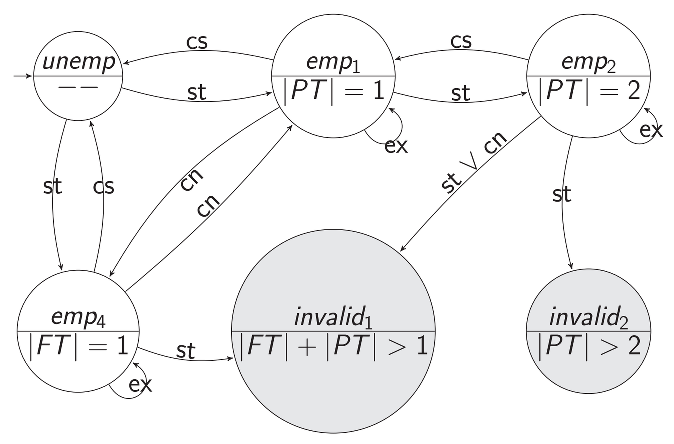

Table: A "one step iteration" Double Check Matrix (with additional information) computed on the careers data of an Italian Area.

	Conditions			Result
Case	*ccheck*(S_i)	*equals*(S_i, C_i)	*ccheck*(C_i)	Cardinality
1	0	0	0	96,353
2	0	0	1	0
3	0	1	0	32,789
4	0	1	1	1,399
5	1	0	0	3
6	1	0	1	40
7	1	1	0	74,904
8	1	1	1	8,078

Conclusion and expected achievements

- RDQA: a Model Checking based approach to evaluate and improve database data quality;
- Exploit Model Checking supporting the generation of *data cleansing* routines.