Comp. by: PG0971 Stage: Revises1 ChapterID: 000000A821 Date:13/6/09 Time:22:01:35

Consistent Facts

► Possible Answers

Consistent Query Answering

LEOPOLDO BERTOSSI
Carleton University, Ottawa, ON, Canada

Definition

Consistent query answering (CQA) is the problem of querying a database that is inconsistent, i.e., that fails to satisfy certain integrity constraints, in such a way that the answers returned by the database are consistent with those integrity constraints. This problem involves a characterization of the semantically correct or consistent answers to queries in an inconsistent database.

Key Points

Databases may be inconsistent in the sense that certain desirable integrity constraints (ICs) are not satisfied. However, it may be necessary to still use the database, because it contains useful information, and, most likely, most of the data is still consistent, in some sense. CQA, as introduced in [1], deals with two problems. First, with the logical characterization of the portions of data that are consistent in the inconsistent database. Secondly, with developing computational mechanisms for retrieving the consistent data. In particular, when queries are posed to the database, one would expect to obtain as answers only those answers that are semantically correct, i.e., that are consistent with the ICs that are violated by the database as a whole.

The consistent data in the database is characterized [1] as the data that is invariant under all the database instances that can be obtained after making minimal changes in the original instance with the purpose of restoring consistency. These instances are the so-called (minimal) *repairs*. In consequence, what is consistently true in the database is what is *certain*, i.e., true in the collection of possible worlds formed by the repairs. Depending on the queries and ICs, there are different algorithms for computing consistent answers. Usually, the original query is transformed into a new query, possibly written in a different language, to be posed to the database at hand, in such a way that the usual answers to the latter are the consistent answers to the former [1]. For surveys of CQA and specific references, c.f. [2,3].

Cross-references

► Database Repairs
► Inconsistent Databases

Recommended Reading

1. Arenas M., Bertossi L., and Chomicki J. Consistent query answers in inconsistent databases. In Proc. 18th ACM SIGACT-SIGMOD-SIGART Symp. on Principles of Database Systems, 1999, pp. 68–79.
2. Bertossi L. Consistent query answering in databases. ACM SIGMOD Rec., 35(2):68–76, 2006.
3. Chomicki J. Consistent query answering: five easy pieces. In Proc. 11th Int. Conf. on Database Theory, 2007, pp. 1–17.

Constant Span

► Fixed Time Span

Constrained Frequent Itemset Mining

► Frequent Itemset Mining with Constraints

Constraint Databases

FLORIS GEERTS
University of Edinburgh, Edinburgh, UK

Definition

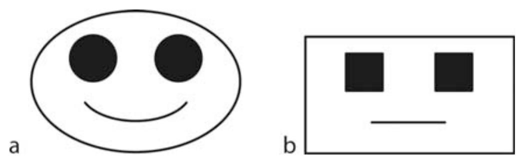
Constraint databases are a generalization of relational databases aimed to store possibly infinite-sized sets of data by means of a finite representation (*constraints*) of that data. In general, constraints are expressed by *quantifier-free first-order formulas* over some fixed vocabulary Ω and are interpreted in some Ω-structure $\mathcal{M} = \langle \mathbb{U}, \Omega \rangle$. By varying Ω and $\mathcal{M}$, constraint databases can model a variety of data models found in practice including traditional relational databases, spatial and spatio-temporal databases, and databases with text fields (strings). More formally, let Ω be a fixed vocabulary consisting of function, predicate and constant symbols, and let $\mathcal{R} = \{R_1,\ldots,R_\ell.\}$ be a relational schema, where each relation name R_i is of arity

$n_i > 0$. An *Ω-constraint database* **D** with schema $\mathcal{R}$ maps each relation $R_i \in \mathcal{R}$ to a quantifier-free formula $\varphi^{\mathbf{D}}_{R_i}(x_1,\ldots,x_{n_i})$ (with n_i free variables $x_1,\ldots,x_{n_i}$) in *first-order logic* over Ω. When interpreted over an Ω-structure $\mathcal{M} = \langle \mathbb{U}, \Omega \rangle$, an Ω-constraint database **D** with schema $\mathcal{R}$ corresponds to the collection of the *$\mathcal{M}$-definable sets* $[\![R_i]\!]^{\mathbf{D}}_{\mathcal{M}} = \{(a_1,\ldots,a_{n_i}) \in \mathbb{U}^{n_i} \mid \mathcal{M} \models \varphi^{\mathbf{D}}_{R_i}(a_1,\ldots,a_{n_i})\}$, for $R_i \in \mathcal{R}$. Constraint query languages have been devised to manipulate and query constraint databases.

Key Points

The primary motivation for constraint databases comes from the field of spatial and spatio-temporal databases where one wants to store an infinite set of points in the real Euclidean space and query it as if all (infinitely) many points are present [3,4,5]. In the spatial context, the constraints used to finitely represent data are *Boolean combinations of polynomial inequalities*. For instance, the infinite set of points in the real plane $\mathbb{R}^2$ depicted in Fig. 1(a) can be described by means of a disjunction of polynomial inequalities with integer coefficients as follows: $\varphi(x, y) = (x^2/25 + y^2/16 = 1) \vee (x^2 + 4x + y^2 - 2y \leq 4) \vee (x^2 - 4x + y^2 - 2y \leq -4) \vee (x^2 + y^2 - 2y = 8 \wedge y < -1)$. In the language of constraint databases, $\varphi(x, y)$ is a quantifier-free first-order formula over $\Omega = (+,\cdot,0,1,<)$ and Fig. 1(a) represents the $\mathcal{M}$-definable set in $\mathbb{R}^2$ corresponding to the formula φ for the Ω-structure $\mathcal{M} = \langle \mathbb{R}, \Omega \rangle$. If $\mathcal{R}$ is a relational schema consisting of a binary relation R, then the Ω-constraint database **D** with schema $\mathcal{R}$ defined by $R \mapsto \varphi(x, y)$ "stores" the set in Fig. 1(a). In this case, the $\mathcal{M}$-definable sets are also known as semi-algebraic sets [2].

When *Boolean combination of linear inequalities* suffice, such as in geographical information systems (GIS), one considers constraint databases over $\Omega = (+,0,1,<)$ and $\mathcal{M} = \langle \mathbb{R}, \Omega \rangle$. Fig. 1(b) shows an example of a set defined by means of a first-order formula over $\Omega = (+,0,1,<)$. The advantage of the constraint approach to represent spatial data is the uniform representation of the various spatial entities. Whereas in GIS one normally defines a special data-type for each spatial object such as line, poly-line, circle,..., each of those are now represented by constraints in the same constraint language.



Constraint Databases. Figure 1. Example of set definable by (a) polynomial constraints and (b) linear constraints.

Other common scenarios of constraint databases include: *dense order constraints over the rationals*, where $\Omega = (<,(c)_{c\in\mathbb{Q}})$ and $\mathcal{M} = \langle \mathbb{Q}, \Omega \rangle$. That is, rational numbers with order and constants for every $c \in \mathbb{Q}$; and *constraints over strings*, where $\Omega = ((f_a)_{a\in\Sigma},\prec,\text{el})$ and $\mathcal{M} = \langle \Sigma^*, \Omega \rangle$ [1]. Here, Σ is a finite alphabet, f_a is a function that adds a at the end of its argument, $\prec$ is the prefix relation and $\text{el}(x, y)$ is a binary predicate that holds if $|x| = |y|$, where $|\cdot|$ stands for the length of a finite string. In the latter case, the $\mathcal{M}$-definable sets are precisely the regular languages over Σ.

Finally, standard relational databases with schema $\mathcal{R}$ can be considered as constraint databases over *equality constraints over an arbitrary infinite domain* $\mathbb{U}$, where $\Omega = ((c)_{c\in\mathbb{U}})$ and $\mathcal{M} = \langle \mathbb{U}, \Omega \rangle$. Indeed, consider a tuple $t = (a_1,\ldots,a_n)$ consisting of some constants $a_i \in \mathbb{U}$, for $i \in [1,n]$. The tuple t can be expressed by the formula $\varphi_t(x_1,\ldots,x_n) = (x_1 = a_1) \wedge \ldots \wedge (x_n = a_n)$ over the signature $\Omega = ((c)_{c\in\mathbb{U}})$. More generally, an instance $I = \{t_1,\ldots,t_N\}$ over $R \in \mathcal{R}$ corresponds to $\varphi_I = \bigvee_{i=1}^{N} \varphi_{t_i}$. Therefore, a relational instance $(I_1,\ldots,I_\ell)$ over $\mathcal{R}$ can be represented as the constraint database **D** defined by $R_i \mapsto \varphi_{I_i}(x_1,\ldots,x_{n_i})$, for $i \in [1,\ell]$. This shows that constraint databases indeed generalize standard relational databases.

Cross-references

► Constraint query languages
► Geographic information system
► Relational model
► Spatial data type

Recommended Reading

1. Benedikt M., Libkin L., Schwentick T., and Segoufin L. Definable relations and first-order query languages over strings. J. ACM, 50(5):694–751, 2003.
2. Bochnak J., Coste M., and Roy M.F. Real Algebraic Geometry. Springer, Berlin, 1998.
3. Kanellakis P.C., Kuper G.M., and Revesz P.Z. Constraint query languages. J. Comput. Syst. Sci., 51(1):26–52, 1995.
4. Kuper G.M., Libkin L., and Paredaens J. (eds.) Constraint databases. Springer, Berlin, 2000.
5. Revesz P.Z. Introduction to Constraint Databases. Springer, Berlin, 2002.