BP-Completeness

Dirk Van Gucht
Computer Science Department
Indiana University

SYNONYMS

Instance-completeness; Relation-completeness

DEFINITION

A relational query language Q is BP-complete if for each relational database D, the set of all relations defined by the queries of Q on D is equal to the set of all first-order definable relations over D. More formally, fix some infinite universe $\mathbf{U}$ of atomic data elements. A *relational database schema* $\mathcal{S}$ is a finite set of relation names, each with an associated arity. A *relational database* D with schema $\mathcal{S}$ assigns to each relation name of $\mathcal{S}$ a *finite* relation over $\mathbf{U}$ of its arity. The domain of D, $dom(D)$, is the set of all atomic data elements occurring in the tuples of its relations. Let $FO^{\mathcal{S}}$ be the set of first-order formulas over signature $\mathcal{S}$ and the equality predicate, and let $FO^{\mathcal{S}}(D) = \{\varphi(D) \mid \varphi \in FO^{\mathcal{S}}\}$. (For a formula $\varphi \in FO^{\mathcal{S}}$ with free variables $(x_1, \ldots, x_m)$, $\varphi(D)$ denotes the m-ary relation over $dom(D)$ defined by φ, where the variables in φ are assumed to range over $dom(D)$.) Let $Q^{\mathcal{S}}$ denote those queries of Q defined over schema S, and let $Q^{\mathcal{S}}(D) = \{q(D) \mid q \in Q^{\mathcal{S}}\}$, i.e., the set of relations defined by queries of Q applied to D. Then, Q is *BP- complete* if for each relational database D over schema $\mathcal{S}$,

$$Q^{\mathcal{S}}(D) = FO^{\mathcal{S}}(D).$$

In the words of Chandra and Harel, "BP-completeness can be seen to be a measure of the power of a language to express relations and *not* of its power to express functions having relations as outputs, i.e., queries." In fact, there exist BP-complete languages that do not express the same queries.

MAIN TEXT

Chandra and Harel introduced the concept of BP-completeness and attributed it to Bancilhon and Paredaens who were the first to study it. Bancilhon and Paredaens considered the following decision problem: given a relational database D and a relation R defined over $dom(D)$, does there exists a first-order formula φ such that $\varphi(D) = R$?[1] They gave an algebraic, language-independent characterization of this problem by showing that such a first-order formula exists if and only if for each bijection $h : dom(\mathbf{D}) \rightarrow dom(D)$, if $h(D) = D$ then $h(R) = R$. Equivalently, if h is an automorphism of D then it is also an automorphism of R. (Here, $h(D)$ and $h(R)$ are the natural extensions of h to D and R, respectively.) For a relational database D over schema $\mathcal{S}$, denote by $Aut(D)$ and $Aut(R)$ the set of automorphisms of $dom(D)$ and D, respectively, and let $R^{\mathcal{S}}(D) = \{R \mid R \text{ is a relation over } dom(D) \text{ such that } Aut(D) \subseteq Aut(R)\}$. Then, an alternative characterization for the BP-completeness of Q is to require that for each relational database D over schema $\mathcal{S}$,

$$Q^{\mathcal{S}}(D) = R^{\mathcal{S}}(D).$$

Van den Bussche showed that this characterization follows from Beth's Theorem about the explicit and implicit definability of first-order logic. The concept of BP-completeness has been generalized as well as specialized to query languages over other database models.

CROSS REFERENCE*

[1] Paredaens considered this problem for the relational algebra, but by Codd's theorem on the equivalence of first-order logic and the relational algebra, these decision problems are the same.

Complete query languages, query language, relational calculus and algebra.

REFERENCES*

F. Bancilhon: On the Completeness of Query Languages for Relational Databases. Proc. Mathematical Foundations of Computer Science 1978.

A. Chandra and D. Harel: Computable Queries for Relational Databases. *Journal of Computer and Systems Science*, 25:99-128, 1982.

J. Paredaens: On the Expressive Power of the Relational Algebra. *Information Processing Letters*, 7(2):107-111, 1978.

J. Van den Bussche: Applications of Alfred Tarski's Ideas in Database Theory. In *Computer Science Logic*, *Lecture Notes in Computer Science*, 2142: 20-37, 2001.