Temporal Relational Calculus

Jan Chomicki, University at Buffalo, USA, http://www.cse.buffalo.edu/~chomicki
David Toman, University of Waterloo, Canada, http://www.cs.uwaterloo.ca/~david

SYNONYMS

two-sorted first-order logic

DEFINITION

Temporal Relational Calculus (TRC) is a temporal query language extending the relational calculus. In addition to data variables and quantifiers ranging over a data domain (a universe of *uninterpreted constants*), temporal relational calculus allows *temporal variables* and quantifiers ranging over an appropriate time domain [1].

MAIN TEXT

A natural temporal extension of the relational calculus allows explicit variables and quantification over a given time domain, in addition to the variables and quantifiers over a data domain of uninterpreted constants. The language is simply the two-sorted version (variables and constants are temporal or non-temporal) of first-order logic over a data domain D and a time domain T.

The syntax of the two-sorted first-order language over a database schema $\rho = \{R_1, \ldots, R_k\}$ is defined by the grammar rule:

$$Q ::= R(t_i, x_{i_1}, \ldots, x_{i_k}) \mid t_i < t_j \mid x_i = x_j \mid Q \wedge Q \mid \neg Q \mid \exists x_i.Q \mid \exists t_i.Q$$

In the grammar, t_i's are used to denote temporal variables and x_i's to denote data (non-temporal) variables. The atomic formulae $t_i < t_j$ provide means to refer to the underlying ordering of the time domain. Note that the schema ρ contains schemas of *timestamped temporal relations* (see the entry Point-stamped Temporal Models).

Given a point-timestamped database DB and a two-sorted valuation θ, the semantics of a TRC query Q is defined in the standard way (similarly to the semantics of relational calculus) using the *satisfaction relation* $DB, \theta \models Q$:

$$\begin{array}{ll}
DB, \theta \models R_j(t_i, x_{i_1}, \ldots, x_{i_k}) & \text{if } R_j \in \rho \text{ and } (\theta(t_i), \theta(x_{i_1}), \ldots, \theta(x_{i_k})) \in R_j^{DB} \\
DB, \theta \models t_i < t_j & \text{if } \theta(t_i) < \theta(t_j) \\
DB, \theta \models x_i = x_j & \text{if } \theta(x_i) = \theta(x_j) \\
DB, \theta \models Q_1 \wedge Q_2 & \text{if } DB, \theta \models Q_1 \text{ and } DB, \theta \models Q_2 \\
DB, \theta \models \neg Q_1 & \text{if not } DB, \theta \models Q_1 \\
DB, \theta \models \exists t_i.Q_1 & \text{if there is } s \in T \text{ such that } DB, \theta[t_i \mapsto s] \models Q_1 \\
DB, \theta \models \exists x_i.Q_1 & \text{if there is } a \in D \text{ such that } DB, \theta[x_i \mapsto a] \models Q_1
\end{array}$$

where R_j^{DB} is the interpretation of the predicate symbol R_j in the database DB.

The *answer to a query Q over DB* is the set $Q(DB)$ of valuations that make Q true in DB. Namely, $Q(DB) := \{\theta_{|FV(Q)} : DB, \theta \models Q\}$ where $\theta_{|FV(Q)}$ is the restriction of the valuation θ to the free variables of Q.

In many cases, the definition of TRC imposes additional restrictions on valid TRC queries:

Restrictions on free variables: often the number of free temporal variables in TRC queries can be restricted to guarantee closure over the underlying data model (e.g., a single-dimensional timestamp data model or the bitemporal model). Note that this restriction applies only to queries, not to subformulas of queries.

Range restrictions: another common restriction is to require queries to be range restricted to guarantee domain independence. In the case of TRC (and many other abstract query languages), these restrictions depend crucially on the chosen *concrete encoding* of temporal databases (see the entry Abstract and Concrete Temporal Query Languages). For example, no range restrictions are needed for temporal variables when queries are evaluated over interval-based database encodings, because the complement of an interval can be finitely represented by intervals.

The schemas of atomic relations, $R_j(t_i, x_{i_1}, \ldots, x_{i_k})$, typically contain a single temporal attribute/variable, often in fixed (e.g., first) position: this arrangement simply reflects the choice of the underlying temporal data model to be the single-dimensional valid time model. However, TRC can be similarly defined for multidimensional temporal data models (such as the bitemporal model) or for models without a predefined number of temporal attributes by appropriately modifying or relaxing the requirements on the structure of relation schemas (see the entry Point-stamped Temporal Models).

An interesting observation is that a variant of TRC, in which temporal variables range over *intervals* and that utilizes *Allen's interval relations* as basic comparisons between interval values, is equivalent to TRC over two-dimensional temporal relations, with the two temporal attributes standing for interval endpoints.

CROSS REFERENCE

abstract and concrete temporal query languages, point-stamped temporal data models, relational calculus, relational model, temporal logic in query languages, temporal query languages, temporal relation, time domain, time instant, TSQL2, valid time.

RECOMMENDED READING

[1] J. Chomicki and D. Toman. Temporal Databases. In M. Fischer, D. Gabbay, and L. Villa, editors, *Handbook of Temporal Reasoning in Artificial Intelligence*, pages 429–467. Elsevier *Foundations of Artificial Intelligence*, 2005.