Logic Programming

Database Management

Michael Genesereth
Computer Science Department
Stanford University
Requests:
  \( \text{add}(p) \) is a request to add \( p \) to the database.
  \( \text{del}(p) \) is a request to remove \( p \) from the database.

Simultaneous updates okay
  \( \text{del}(p(a)) \)
  \( \text{del}(q(b)) \)
  \( \text{add}(p(b)) \)
  \( \text{add}(q(a)) \)

NB: DB system may or may not act upon requests. Might reject, might accept as is, might modify.
Suppose we have a database and a set of constraints and suppose the user requests an update which, when applied, produces a dataset that violates the constraints.

Action 1: Reject the update

Action 2: Repair the update
A partition constraint states that every object falls into exactly one of a collection of disjoint classes.

**Ruleset**

\[
\text{add(female}(X)) \implies \neg\text{male}(X) \\
\text{add(male}(X)) \implies \neg\text{female}(X)
\]

**Dataset:** \{male(art), male(chris), female(bea)\}

**Request:** add(female(chris))

**Result:** \{male(art), female(bea), female(chris)\}
A functional dependency states that a relationship has one and only one value for each argument, e.g. father, mother, year in school, etc.

**Ruleset**

\[
\text{add}(\text{year}(S,Y)) \& \text{year}(S,O) \implies \neg \text{year}(S,O) \\
\text{add}(\text{year}(S,Y)) \implies \text{year}(S,Y)
\]

\[
\text{add}(\text{year}(S,Y)) \& \text{year}(S,O) \\
\implies \neg \text{year}(S,O) \& \text{year}(S,Y)
\]

**Dataset:** \{\text{year}(a,1), \text{year}(b,2), \text{year}(c,3)\}

**Request:** add(\text{year}(b,3))

**Result:** \{\text{year}(a,1), \text{year}(b,3), \text{year}(c,3)\}
A inclusion dependency states that, if an object appears as an argument of one predicate, then it must also appear in a specific position in another predicate.

**Ruleset**
\[ \text{add}(p(X,Y)) \implies \text{adult}(X) \]

**Dataset:** \{adult(art), p(art,bea)\}

**Request:** add(p(bea,coe))

**Result:**
\{adult(art), adult(bea), p(art,bob), p(bea,coe)\}
A **materialized view** is a view relation that is stored explicitly in the database.

**Ruleset**

\[
father(X,Y) :\neg \text{parent}(X,Y) \& \text{male}(X)
\]
add(parent(X,Y)) & male(X) & ~del(male(X))
    ==> father(X,Y)

parent(X,Y) & add(male(X)) & ~del(parent(X,Y))
    ==> father(X,Y)

add(parent(X,Y)) & add(male(X))
    ==> father(X,Y)

del(parent(X,Y)) ==> ~father(X,Y)

del(male(X)) ==> ~father(X,Y)
View Definition

\[ r(X) :- p(X) \]
\[ r(X) :- q(X) \]

Request: \texttt{add}(r(bob))

Positive Update:

\{p(bob)\}?
\{q(bob)\}?
\{p(bob), q(bob)\}? \text{ What if } p \text{ is male and } q \text{ is female?}
\{\}\?
View Definition

\[ r(X) :- p(X) \]
\[ r(X) :- q(X) \]

Update Rule

\[ \text{add}(r(X)) \land \neg q(X) \implies p(X) \]