

Diss. ETH No. 10802

Managing Temporal Knowledge in Deductive Databases

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZÜRICH

for the degree of
Doctor of Technical Sciences

presented by
MICHAEL BÖHLEN
Dipl. Informatik-Ing. ETH
born 8. October, 1964
citizen of Riggisberg (BE)

accepted on the recommendation of
Prof. Dr. R. Marti, examiner
Prof. Dr. H.-J. Schek, co-examiner

1994

Abstract

Many applications have to deal with a large amount of data which not only represent the perceived state of the real world at present, but also past and/or future states. This type of time-varying data arises for example in banking and insurance applications, in health care, in planning and scheduling problems, and in ticket reservation systems. These applications are not served adequately by today's database systems. In particular, deletions and updates in such systems have destructive semantics. This means that previous database contents (representing previous perceived states of the real world) cannot be accessed anymore. Moreover, the formulation of queries which access different database states is often tedious and hence error prone without special support for the temporal dimension.

Temporal database systems have been developed to provide this kind of support. They assume that all data values are associated with timestamps which represent the time during which this data value was valid in the real world and/or the time during which this data value was recorded in the database.

The work presented in this thesis generalizes temporal extensions proposed for relational databases and languages (e.g., TQuel and TSQL) to deductive databases. In contrast to previous approaches in the area of logic programming, our approach also lends itself nicely to an efficient implementation of a deductive database system which provides all typical database services such as persistence, concurrency control, recovery etc.

We have identified the notion of temporal irreducibility as the key to ensure the correctness of answers to temporal queries. From an abstract point of view, temporal irreducibility requires that there is no redundancy in the timestamps associated with identical data values. Moreover, temporal irreducibility is a cornerstone of the definition of a new notion of completeness for temporal query languages.

On a practical level, we have designed and implemented the temporal deductive database language ChronoLog which is based on Datalog and on the above mentioned completeness notion. ChronoLog departs from languages such as TQuel and TSQL in that timestamps associated with derived tuples are computed implicitly. This allows a concise formulation of most types of temporal queries. Our implementation relies on a translation of ChronoLog statements to SQL statements which are in turn executed by a commercial relational database manager.

Zusammenfassung

Viele Anwendungen müssen grosse Datenmengen, die nicht nur den momentan gültigen Zustand sondern auch frühere und/oder zukünftige Zustände der realen Welt beschreiben, handhaben. Diese Art von zeitbezogenen Daten entsteht beispielsweise bei Bank- und Versicherungsanwendungen, bei Krankengeschichten, bei Planungsproblemen sowie bei Reservationssystemen. Anwendungen dieser Art werden von heutigen Datenbanksystemen nicht angemessen unterstützt. Insbesondere haben in solchen Systemen Änderungs- und Löschoptionen eine destruktive Semantik. Damit kann nicht mehr auf frühere Datenbankinhalte (welche frühere Zustände der realen Welt modellieren) zugegriffen werden. Ausserdem ist die Formulierung von Anfragen, welche auf unterschiedliche Datenbankzustände zugreifen, mühsam und somit fehleranfällig.

Temporale Datenbanksysteme wurden entwickelt um diese Art von Anwendungen besser zu unterstützen. Sie gehen davon aus, dass alle Daten mit einer Zeitmarke versehen sind, die die Zeit darstellt während der ein Datenwert in der realen Welt gültig war und/oder in der Datenbank sichtbar war.

Die vorliegende Arbeit generalisiert zeitliche Erweiterungen von relationalen Datenbanken und Sprachen (zum Beispiel TQuel und TSQL) zu deduktiven Datenbanken. Im Gegensatz zu anderen Ansätzen aus dem Gebiet der logischen Programmierung eignet sich unser Ansatz als Grundlage für eine effiziente Implementation eines deduktiven Datenbanksystems, welches alle Eigenschaften einer Datenbank wie etwa Persistenz, Mehrbenutzerbetrieb und Datenwiederherstellung (*recovery*) bewahrt.

Wir haben den Begriff der temporalen Unreduzierbarkeit als Schlüssel zur Korrektheit der Beantwortung zeitlicher Abfragen identifiziert. Aus einem abstrakten Blickwinkel heraus betrachtet verlangt die temporale Unreduzierbarkeit, dass die Datenwerte mit den zugehörigen Zeitmarken redundanzfrei gespeichert sind. Ausserdem stellt die temporale Unreduzierbarkeit einen Grundstein für eine neue Definition der Vollständigkeit temporaler Abfragesprachen dar.

Auf der praktischen Seite habe wir ChronoLog, eine temporale deduktive Abfragesprache, die auf Datalog und dem vorgängig erwähnten Vollständigkeitsbegriff aufbaut, entworfen und implementiert. ChronoLog hebt sich von Sprachen wie TQuel und TSQL dadurch ab, dass die Zeitmarken abgeleiteter Daten implizit berechnet werden. Dies ermöglicht es, die meisten zeitbezogenen Abfragen prägnant zu formulieren. Unsere Implementation basiert auf einer Übersetzung von ChronoLog-Befehlen nach SQL-Befehlen, die schliesslich durch ein kommerzielles relationales Datenbanksystem ausgeführt werden.