Future of Deductive Database

Dr. Lamyaa A.S. Al-Sammarae
Computer Science Department
Al_Mustansiriya University

2006
**Future of Deductive Database**

1. **Abstract**

This Paper presents and identifies number of factors that will help to make a view to the future of deductive data bases to make such view this require to know major development in this field to assist specialists who will need to make use of the significant developments in deductive databases. The efficient handling of recursion and the recognition that some recursive cases might inherently be bounded contributes to the practical implementation of Deductive Databases.

This paper identifies as well factors that effect the area of implementations of Deductive databases Status of Deductive Databases, where commercial developments have not progressed as rapidly as intellectual developments. Some of the these significant developments will rapid the progressed of Deductive databases presents the main factors that may influence the future deductive data base applications.

2. **Background**

The use of logic and deduction in databases, started in the late 1960s. Prominent among developments was work by Levien and Maron (1965) and Kuhns (1967), and by Green and Raphael (1968), who were the first to realize the importance of the Robinson (1965) resolution principle for databases. [1]

A major influence on the use of logic in databases was the development of the field of logic programming: in (1974) the concept of logic as a programming language was promulgated, developed the first Prolog interpreter referred to as logic programs that are function free as deductive databases, or as datalog. [2]

The impetus for the use of logic in databases came at 1976. The book Logic and Data Bases (1978), edited by Gallaire and Minker, was highly influential in the development of the field, as were the two volumes of Advances in Database Theory (Gallaire, Minker, and
Nocholas 1984, 1981). Another influential development was the article by Gallaire, Minker, and Nicolas (1984), which surveyed work in the field to that point.[5]

The use of logic in databases was received by the database community with a great deal of skepticism: Was deductive databases a field? Did DDBs contribute to database theory or practice? The accomplishments in this article are testaments to the fact that logic has contributed significantly both to the theory and the practice of databases. It is clear that logic has everything to do with the theory of databases, and many of those who were then critical of the field have changed their position.

3. Logic in Deductive Databases

Codd (1970) formalized databases in terms of the relational calculus and the relational algebra. He provided a logic language and the relational calculus and described how to compute answers to questions in the relational algebra and the relational calculus. Both the relational calculus and the relational algebra provide declarative formalisms to specify queries. This was a significant advance over network and hierarchic systems (Ullman 1989, 1988), which only provided procedural languages for databases. The relational algebra and the relational calculus permitted individuals who were not computer specialists to write declarative queries and have the computer answer the queries. The development of syntactic optimization techniques (Ullman 1989, 1988) permitted relational database systems to retrieve answers to queries efficiently and compete with network and hierarchic implementations. Relational systems have been enhanced to include views. A view, as used in relational databases, is essentially a non-recursive procedure. There are numerous commercial implementations of relational databases systems for large database manipulation and for personal computers. Relational databases are a forerunner of logic in databases.

Although relational databases used the language of logic in the relational calculus, it was not formalized in terms of logic. The formalization of relational databases in terms of logic and the extensions that have been developed are the focus of this article. Indeed, formalizing databases through logic has played a significant
role in our understanding of what constitutes a database, what is meant by a query, what is meant by an answer to a query, and how databases can be generalized for knowledge bases. It has also provided tools and answers to problems that would have been extremely difficult without the use of logic. [6]

4. Types of DBMS

This section presents the most representative of the existing models in Deductive Database Management Systems DBMS, these are:- LDL its extension LDL++ followed by the NAIL! system [and its combination with the Glue language, then the Coral system and Coral++ as well as a synthesis of other systems such as: XSB, Declare, Logres, and recent projects Aditi & Aditi2 2005.

Let’s take, for example, the LDL++ system which The compiler reads the LDL++ programs and builds the connection graph of the global predicate (GCP). The compiler evaluates, for each query, the GCP partially, transforming the predicate into an object network executed by the interpreter. Thus, the compiler is responsible of checking the rules safety and is also responsible of rewriting the recursive rules using techniques such as magic sets. Besides accessing the internal database, LDL++ can also access several types of external SGBD.

The interpreter receives as entry a graph of the runnable objects equivalent to a LDL++ query generated by the compiler and it runs it by the get-next call and by other calls on the local database. The same thing happens in the external databases as well as in the external predicates system. All the programs written in C/C++ can call up the LDL++ system by a standard API interface, so the programs written in LDL++ could be inserted in other procedural systems. [5]

NAIL (Not Another Implementation of Logic). A language called GLUE [9], developed for logical rules, has the power of SQL statements as well as a conventional language for the construction of loops, procedures, and modules.

Among other significant model and the latest one is the Aditi model:

The ultimate goal of the Aditi project is to show that this generalization does not have to compromise performance. Applications that can run on relational systems should run on deductive systems with similar performance while being substantially
easier to develop, debug and maintain. Performance was kept for traditional applications competitive by using conventional relational technology whenever possible; programming costs kept down by allowing developers to work at a level much higher than SQL and other traditional query languages. At this level, many problems could be solved without resorting to a host language. The higher level also allows the system to perform automatically better global optimization than is currently possible, because the humans who currently do global optimization cannot keep in mind all the relevant details at the same time. This is particularly important for the most sophisticated applications, such as expert systems that work on large amounts of data. Therefore deductive databases can make feasible complex applications that were not feasible before. [ 3 ]

Then Aditi2

A beta version of the second major release of Aditi is now available. The Aditi 2 system is distributed under the terms of the GNU General Public License.

This release has been tested on considerably fewer platforms than previous releases. There are a some limitations and problems in the current release. It is worth noting that Aditi now includes its own build tool, and only uses make to create that. A personalized version is used now. There are is no further development planned for Aditi [ 3 ]

5. **Effective Developments**

The following developments portend, relational databases are starting to incorporate techniques stemming from research in Deductive Data bases.

1. Until recently, no convincing demonstration has been made of a large commercial problem that requires a Deductive DB, which might be why the developments were terminated.

2. However, now, a large number of applications could take advantage of this technology, as evidenced by the applications being performed by the VALIDITY deductive object-oriented database (DOOD) system. In addition, Levy et al. (1995) studied the problem of
computing answers to queries using materialized views and note that this work is related to applications such as global information systems, mobile computing, view adaptation, and the maintenance of physical data independence. Levy et al. (1996) describe how Deductive databases can be used to provide uniform access to a heterogeneous collection of more than 100 information sources on the World Wide Web. [1]

3. As noted in Melton (1996): "The use of the recursive union operator allows both linear (single-parent, or tree) recursion and nonlinear (multi parent, or general directed graph) recursion. This solution will allow easy solutions to bill-of-material and similar applications."

4. Linear recursion is currently a part of the client server of IBM's DB2 system. IBM is using the magic set method to perform linear recursion. Also, indications are that the ORACLE database system will support some form of recursion.

5. Another technology available for commercial use is cooperative databases. The tools and techniques exist, as evidenced by COBASE (Chu, Chen, and Merzbacher 1994) and CARMIN (Gaasterland et al. 1992). With the introduction of recursion techniques into relational database technology, it will be necessary to provide users with cooperative database responses so they understand why certain queries fail or succeed. It will also permit queries to be relaxed when the original query fails, permitting reasonable, if not logically correct, answers to be provided to users. Because user constraints can be handled in the same way that integrity constraints are handled.

6. Two significant developments have taken place in the implementation of commercial Deductive DBs. First is the incorporation of techniques developed in Deductive databases into relational technology. Recursive views that use technique for implementation are being permitted, and methods developed for SQO are being applied. Second is the development of a Deductive Object Oriented Database DOOD, VALIDITY, that is in commercial use technology to simulate the capabilities of DDB systems. [9]
7. Programming can be used to form inductive inferences, and Knowledge Discovery in Databases

8. Knowledge bases are important for AI and expert system developments. A general way to represent knowledge bases is through logic. Work developed for extended Deductive Databases concerning semantics and complexity applies directly to knowledge bases. Baral and Gelfond (1994) describe how extended DDBs can be used to represent knowledge bases.

6. Conclusion

- A deductive database system is a database system which can make deductions (i.e., infer additional rules or facts) based on facts stored in the (deductive) database of an expert system or other database systems.
- Observation on major effective development and factors effect the area of implementations of Deductive databases in Implementation Status of Deductive Databases, where commercial developments have not progressed as rapidly as intellectual developments.
- Some of these significant developments will rapid the progressed of Deductive databases, the efficient handling of recursion and the recognition that some recursive cases might inherently be bounded contributes to the practical implementation of Deductive databases.
- The (8) factors specified in “Effective Development” in this research would help to make a thorough idea of the future of this field.

7. References


2. Asirelli, P.; Renso, C.; and Turini, F” Language Extensions for Semantic Integration of Deductive Databases (LID), . 1996


14. J. Rauch, “Logical Calculi for Knowledge Discovery in Databases”, 1997,

15. Jiawei Han “Constraint-Based Query Evaluation in Deductive Databases”
مستقبل قواعد البيانات الاستنتاجية

د. لميا عبد الصمد السامرائي
قسم علوم الحاسبات/ كلية العلوم/ الجامعة المستنصرية

يوجد هذا البحث ويحدد عدد من العوامل التي سوف تساعد على تكوين صورة واضحة عن مستقبل قواعد البيانات الاستنتاجية Deductive Databases ان المعالجة الكفوية لعملية التكرار الذاتي Recursion التي وفرت هذه القواعد ومعرفة ان بعض حالات التكرار الذاتي يمكن أن تكون محددة ومقيدة بنوع التطبيقات لقواعد البيانات الاستنتاجية ويتطلب ذلك معرفة أهم التطورات التي حصلت في هذا الحقل حيث ان معرفته تساعده المختصين على الاستفادة من هذه التطورات وتشخيص العوامل التي اثرت وأدت إليها. كما يكتشف العوامل التي تؤثر في مجال تطبيقات قواعد البيانات الاستنتاجية حيث لا يكون التطور في التطبيق التجارية سريعا كسرعته في التطبيقات الفكرية. ان بعض هذه التطورات سوف تعجل من تقدم قواعد البيانات الاستنتاجية محددا اهم العوامل التي تؤثر على مستقبل التطبيق لهذه القواعد.