SCHEMA HYPERGRAPHS: A FORMALISM TO INVESTIGATE LOGICAL
DATA BASE DESIGN

Carlo BATINI - Alessandro D'ATRI
Istituto di Automatica - Università di Roma e CSSCCA - CNR
Via Eudossiana 18 - 00184 - ROMA - Italy

ABSTRACT

In this paper we apply the formalism of hypergraphs and hypergraph
context free grammars to characterize and investigate various aspects of
logical data base design.

We show that the hypergraph formalism is a uniform notation to re-
present data and properties of data. Several results concerning context
free generation of conceptual schemata are proved.
1. INTRODUCTION

In the recent literature, the process of database design is divided in five steps (see [7]):
1) Corporate requirement Analysis; 2) View Modelling; 3) View Integration;
4) Implementation Design; 5) Physical Data Base Design.

The result of the first and second step is an abstract representation of each application view (or user conceptual schema) of interest for the enterprise.

During the third step the different application views are integrated into one global or enterprise view or global conceptual schema of the data base (Second and third step are usually called Logical Data Base Design).

Finally steps 4 and 5 provide the physical implementation of the data base.

In this paper we investigate various aspects of the view modelling step of logical data base design.

We assume in the following as abstract representation of an application view the n-ary relational model of data (see Appendix for the basic definitions we use in this paper).

In the n-ary relational model an application view is a set of relation schemata together with a set of data dependencies among attributes. In this paper it is shown how views can be represented by a notation based on a special kind of hypergraph, i.e. - schema hypergraphs.

In section 2 we introduce the formal machinery and show how the hypergraphs are a suitable and unifying tool of description for various heterogeneous aspects of the relational model of data.

In former papers several formal tools for data base design have been investigated, this process is seen in [2],[3]as a grammar, and the steps of the design of a data base application as a set of refinement step, obtained by activating derivation rules of the grammar.

Suitable restrictions have been investigated in the hypergraph grammar