Chapter 5

CONSISTENCY AND SERIALIZABILITY
IN CONCURRENT DATABASE SYSTEMS

D. J. ROSENKRANTZ* AND R. E. STEARNS†

Department of Computer Science, State University of New York at Albany, Albany, NY 12222, USA

P. M. LEWIS II

General Electric Corporation, Research and Development Center, Schenectady, NY 12301, USA


Abstract

The main results of this paper show that serialization is both necessary and sufficient for consistency in concurrent database systems. This is true for both the final database and the views of the database seen by individual transactions. The model of a transaction includes both read and write operations which may be performed in any order (except an entity must be read before being written).

The main results are presented in terms of an information flow model describing the source of each value read and the use of each value written. Since the model does not involve any concept of the “time” a value was read or written, it models any concurrency system producing information flow among transactions.

There is a section discussing the effect of changing the model to include write operations without preceding reads, and a section discussing the restriction to straight-line programs.

Keywords: database, concurrency control, consistency, serialization, transaction

* The research of this author was supported in part by the National Science Foundation under grant MCS 78-03157.
† The research of this author was supported in part by the National Science Foundation under grant MCS 79-03770.

1. Introduction

There has been a lot of activity in the area of database concurrency controls. The goal of concurrency control is to allow transactions accessing a common database to run as concurrently as possible without destroying database consistency or preventing a transaction from eventually running to completion. It has generally been appreciated that consistency can be insured by designing a serializable control, where serializability means that the effect of running transactions concurrently is the same as if the transactions have been run in some serial order. Many practitioners have in fact made serializability a design requirement.

This paper investigates the relationship between serializability and consistency. We first develop a general concurrency control model based on information flow between transactions. We then show that serializability is both necessary and sufficient for consistency. (There is a small loophole for read-only transactions.) We consider both the consistency of the final database produced by the transactions and the consistency of the view seen by each transaction.

Our concurrency model is developed to reflect assumptions we believe appropriate for concurrency controls in mainstream commercial database systems. These assumptions and our reason for making them are as follows:

ASSUMPTION A1. The control can distinguish between a read access and a write access. Reason. Data manipulation languages have this feature.

ASSUMPTION A2. The control does not know the consistency criterion. Reason. In practice, consistency conditions are too complex to expect a user to write them down (or even fully comprehend them).

ASSUMPTION A3. The control does not make inferences from the particular values read or written. Reason. Because of Assumption A2, this information is, for practical purposes, useless. (In theory, inferences could be made from testing values for equality.)

ASSUMPTION A4. The control may respond to a read request with a value other than the last value written. Reason. Concurrency control designs have been proposed which have this feature [1, 2, 13–15].

ASSUMPTION A5. A value written in the database during the run of the transaction must be considered functionally dependent on all values read, rather