

A Review of Theoretical Methodologies
for Locking in a
Concurrent Data Base Environment

by

Alice Ruth Galinat

B.S. Southeastern Massachusetts University, Dartmouth, 1976

A Masters Report

Submitted in partial fulfillment of the

requirements for the degree

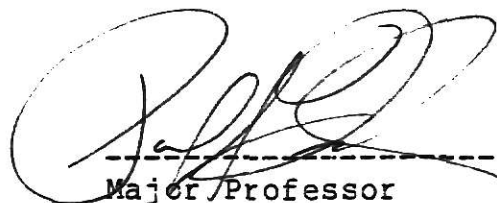
MASTER OF SCIENCE

Department of Computer Science

Kansas State University
Manhattan, Kansas

1980

Approved by:



Major Professor

Spec. Coll.
LD
2668
.R4
1980
G34
c.2

Table of Contents

	Page
List of Tables.....	iii
List of Figures.....	iv
Chapter	
1. Introduction.....	1
2. The Complications of Using a Data Base as a System Resource.....	3
3. Data Base Access, Serial Schedules, and Integrity.....	9
3.1 Data Base Access and Serialization.....	9
3.2 The Added Integrity Problems Caused by Predicate Locks.....	22
3.3 Types of Integrity and Consistency.....	26
4. Locking Protocols.....	30
4.1 Physical Locking Protocols Without Deadlock Prevention.....	31
4.2 Physical Locking Protoccls With Deadlock Prevention.....	37
4.3 Predicate Locking Protocols.....	45
4.4 Concurrent Access to a File of Records.....	53
5. Conclusion.....	58
References.....	66

List of Tables

Table	Page
1. The Compatibility of the Lock Modes in [GRAY75]'s Protocol.....	33
2. The Compatibility of Lock Modes For [BAYE76]'S Protocol.....	45
3. An Extension of Table 1 for [GRAY75]'s Protocol.....	47
4. The Extension of A, IA, and SIA Modes to Table 1.....	47
5. Comparison of Lock Protocols.....	60

List of Figures

Figure	Page
1. Two Simple Transactions.....	15
2. The Dependency Graph of a Simple Non-Serial Schedule.....	15
3. Three Arbitrary Transactions.....	17
4. An Arbitrary Schedule for the Three Transactions in Figure 3.....	18
5. The Dependency Graph for the Arbitrary Schedule in Figure 4.....	19
6. The Schedule and Lock Results of an Arbitrary Schedule Using Predicate Locks.....	23
7. The Weighted Dependency Graph of the Arbitrary Schedule in Figure 6.....	24
8. The Lock Hierarchy Given for the Protocol in [GRAY75].....	31
9. The Ordering of Privileges Between Lock Modes for [GRAY75]'s Protocol.....	34
10. The Partitioning of a File From a Lock Protocol Designed to Handle Predicate Locks [GRAY75].....	35
11. The Schedule of Two Transactions that does not Maintain Integrity in a Weak Sense.....	40
12. The Dependency Graph of the Schedule Given in Figure 11.....	41
13. A Hierarchy of Records for Locking.....	42

14.	An Acceptable Locking Sequence for a Transaction Accessing Records with Figure 13's Hierarchy.....	43
15.	An Example of a B-tree With $N = 4$	53
16.	Key for Table 5.....	61

Chapter 1

Introduction

A data base is a system resource like other system resources that has to be managed by a data base management system (DBMS) or an operating system. Obviously there are differences between a data base and conventional system resources that have to be considered to achieve acceptable levels of resource utilization, throughput, and response time. These differences are discussed in the second chapter. The traditional solutions used for resource allocation are not adequate for a data base, and their problems are discussed. Some of the problems to avoid when creating new solutions are also discussed.

Unlike conventional system resources it is possible to separate a data base into distinct pieces that are treated as separate resources to allow concurrency. This leads to many more problems. The integrity of the data base is threatened by concurrent access. This is the subject of chapter 3. A dependency graph is defined to help study the integrity problems created by concurrent access, and two kinds of integrity are defined from the dependency graph.

The added integrity problems created by using predicate locks instead of physical locks is also discussed. The dependency graph is modified to help study these added problems.

Different levels of integrity are defined at the end of this chapter based on what a transaction (process or user) may do to specified data. The usefulness of the different levels for different kinds of applications is suggested.

Algorithms developed to handle these problems at different integrity levels are presented in chapter 4. The amount of concurrency possible and the overhead involved in each algorithm are discussed. These algorithms are designed for a central data base holding one copy of the data. The supplemental algorithms needed by some locking protocols to detect deadlock and the subject of restart and rollback are not considered.

In the last chapter, chapter 5, some simulation studies concerning the efficiency and concurrency of locking protocols are reviewed in terms of the overhead involved for the various algorithms. The protocols are also compared.