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

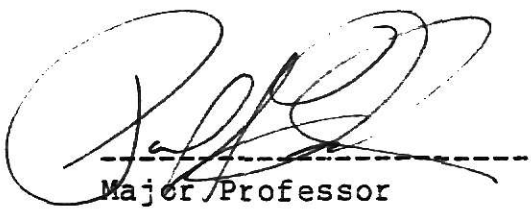
MASTER OF SCIENCE

Department of Computer Science

Kansas State University
Manhattan, Kansas

1980

Approved by:



Major Professor

Table of Contents

Page

List of Tables

Table Page

List of Figures

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.