# ILP:分散式系統之複版控制協定
# Intersect Loops Protocol: A General Protocol for Replica Control in Distributed Systems

王偉筠
Wei-Yun Wang

國立中興大學資訊科學研究所
Institute of Computer Science
National Chung Hsing University
wywang@dblab.cs.nchu.edu.tw

賈坤芳
Kuen-Fang J. Jea

國立中興大學資訊科學研究所
Institute of Computer Science
National Chung Hsing University
jea@dblab.cs.nchu.edu.tw

陳世穎
Shi-Ying Chen

國立臺中商業專科學校
National Taichung Institute of Commerce
sychen@alpha3.ntcic.edu.tw

## 摘要

*分散式系統常用存取門檻的觀念來確保複版控制所有動作的正確性。本文中，我們提出一種新的方法：利用 torus 表面上兩條彼此相交的迴路定義存取門檻；許多已提出的方法可利用這種觀點重新表示。*

關鍵字：分散式系統，複版，容錯，存取門檻

## Abstract

*The idea of read/write quorums is used to ensure the correctness of operations of replica control in distributed systems. In this paper, we propose a novel protocol which uses the idea of two intersecting loops on the torus surface to define quorums. Many proposed protocols can be reformulated by this protocol.*

**Keywords:** distributed systems, replicas, fault tolerance, read/write quorum

## 1. Introduction

A distributed system consists of a collection of autonomous computers linked by a computer network, with software designed to produce an integrated computing facility [4]. Replication is a key to provide good performance, high availability, balanced load sharing, and fault tolerance in distributed systems. Distributed systems must maintain all of the data copies of a data object in different sites. The way to ensure data consistency is that any read and any write operations must satisfy the following constraints: (1) any read operation intersects any write operation, (2) any two write operations intersect. These two constraints guarantee that any two conflicting operations access at least one common data copy.

In this study, we try to find a protocol that is both highly efficient and general. We present a new scheme, which integrates several previously developed protocols and uses the scheme to show the way to control data copies. By analyzing this scheme, it is quite easy to understand the differences among all the developed protocols.

The rest of this paper is organized as follows: Section 2 reviews some related work on replica control protocols. Section 3 describes our proposed intersect loops protocol (ILP) in detail. Section 4 analyzes the protocol and reformulates several previous protocols by ILP. We will illustrate the analysis results of a 4x4 test sample for some reformulated protocols in Section 5. Section 6 is our conclusions with possible future work.

## 2. Related Work

In this section, we first indicate several properties important to the replica control schemes, then we survey some related work. Coterie, read-one-write-all protocol, quorum consensus protocol, grid protocol, and triangular lattice protocol are included.

*Availability* is the probability of a system which is available. Read(write) availability is the probability that the system can perform a read(write) operation. It indicates whether the system is fault-tolerant enough.

*Quorum size* is the number of nodes in a quorum. Large quorum size requires more nodes for the operation. Thus, the system load will increase and the system efficiency will be degraded.

*Load share rate* is the ratio of the load of each node to the load of the node whose load is the minimal in the system. If the load is not balanced, the higher load node must be more powerful.

High availability, small quorum size and balanced load rate usually mean high fault-tolerance ability, high efficiency and no bottleneck, respectively. Unfortunately, the three properties may affect each other. Proposed related protocols are as follows.

*Coterie*[1] is the most general scheme to define quorums. Any kind of protocol that maintains replication is a subset of coterie. A coterie is defined as a set of elements. Each element of a coterie is a set of sites in a distributed system. Two conditions must be held: (1) the intersection property: any two members of a coterie intersect, (2) the minimality property: there are no members of a coterie such that one member contains the

other. We can define both read quorum and write quorum as a set of the members of a coterie.

In *Read-One-Write-All (ROWA) Protocol*, read operations lock only one copy of replicated data, but write operations must lock all of the copies. ROWA has the best read quorum size, read availability, and the worst write quorum size, write availability and write load. Although some protocols[2,3] are proposed to enhance it, they are complex and difficult to implement.

In *Quorum Consensus Protocol*, each copy of the replicated data object is assigned a positive weight. The read(write) quorum is defined as a set of data copies such that the total weight of these data copies is at least RT(WT). To ensure one-copy serializability, RT and WT should satisfy the following constraints: (1) RT + WT > total weights of the data copies. (2) 2*WT > total weights of the data copies.

In Grid Protocol[5], the data copies are organized as an m×n grid. The read quorum is defined as a set of nodes that contains one node from each column or all nodes from one column of the grid. The write quorum is defined as a set of nodes that contains one node from each column and all nodes from one column of the grid.

In *Triangular Lattice Protocol (TLP)* [7,8], the data copies are organized as an m×n triangular lattice. A horizontal crossing is defined as a set of nodes constructing a path connecting the left and right sides, while a vertical crossing connects the top and bottom sides. The read quorum is defined as a set of nodes that contains either a vertical crossing or a horizontal crossing. The write quorum is defined as a set of nodes that contains both a vertical crossing and a horizontal crossing. In Figure 2.1, {1,6,7,12} and {1,2,7,8,11,15} are examples of read and write quorums, respectively.

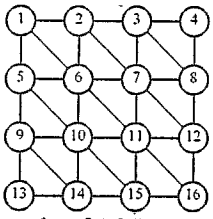


**Figure 2.1** An Example of 16 Data Copies Organized as a 4x4 Triangular Lattice

## 3. The Intersect Loops Protocol

In this section, we propose our new scheme that can integrate several previously proposed replica control protocols.

### 3.1 Intersect Graph

Assume there is a rectangle on a plane and some nodes are put on or in this rectangle. Nodes are linked by two sets of links. One set of the links forms paths that connect the top and the bottom boundaries, while the other forms paths connecting the left and the right boundaries. Two distinct links, each from one of the two sets, can not cross each other. Two distinct paths, each is formed by one of the two sets, can not cross each other on links, but must intersect at one or more node.

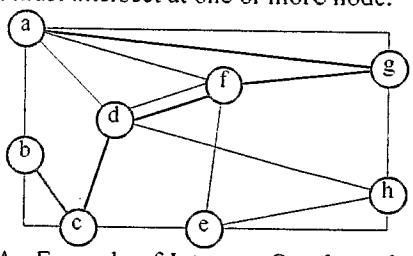


**Figure 3.1** An Example of Intersect Graph on the Plane Structure

In Figure 3.1, the dotted lines are boundaries. Two sets of links are drawn differently by dark and light lines. The dark links do not cross the light links. There are at least one set of dark links forms a path which connects the top and the bottom boundaries and at least one set of light links forms a path which connects the left and the right boundaries. For example, path *b-c-d-f-g* connects the left and the right boundaries. Path *a-d-h-e* connects the top and the bottom boundaries. Path *b-c-d-f-g* and *a-d-h-e* intersect at node *d*. Later, we will apply the concept to torus structures.

***Definition 3.1*** An **intersect graph on a plane** is a graph on a plane which contains nodes and two sets of links, namely *H_type* links and *V_type* links. H_type links form the paths in the horizontal direction and V_type links form the paths in the vertical direction. Two distinct links, each from one of the two sets, can not cross each other.

***Definition 3.2*** An **atomic path** is a path which does not contain any loop.

***Definition 3.3*** An **H_type (V_type) path** is an atomic path composed of H_type (V_type) links and connecting the top and the bottom ( the left and the right) boundaries.

### 3.2 Intersection Property

#### 3.2.1 On Plane Structure

For an intersect graph on a plane structure, the nodes are located on a plane that has four boundaries: top, bottom, right and left boundaries. H_type and V_type links connect all of the nodes in the graph.

***Theorem 3.1*** Any H_type path and V_type path must intersect at one or more node in the intersect graph on the plane structure.

#### 3.2.2. On Torus Structure

An example of the torus structure is shown in Figure 3.2. On a torus structure, the nodes in the

intersect graph are placed on the surface of a torus. All of the nodes are connected by H_type and V_type links.

**Figure 3.2** Torus Structure

***Definition 3.4*** An **intersect graph on a torus** is a graph which contains nodes and two sets of links, namely *H_type* links and *V_type* links. H_type links form the loops in the horizontal direction and V_type links form the loops in the vertical direction. Two distinct links, each from one of the two sets, can not cross each other.

***Definition 3.5*** An **atomic loop** is a loop which does not contain other loops.

***Definition 3.6*** An **H_type (V_type) loop** is an atomic loop composed of H_type (V_type) links.

***Definition 3.7*** A **simple vertical (horizontal) circle** is a circle that rounds the torus once in the vertical (horizontal) direction.

***Lemma 3.1*** An H_type loop crosses any simple vertical circle on the torus. Also, a V_type loop crosses any simple horizontal circle on the torus.

***Theorem 3.2*** Any two of the H_type loop and the V_type loop must intersect at one or more node in the intersect graph on the torus structure.

Because a plane structure can be included by a torus structure, we will use the torus structure to describe our intersect loops protocol in the next section.

### 3.3 The Intersect Loops Protocol (ILP)

***Definition 3.8*** An *R_type loop* is a loop which belongs to any one of the H_type or V_type loops in the intersect graph on the torus structure. A *W_type loop* is a loop which combines two distinct loops from the H_type and V_type loops in the intersect graph on the torus structure.

In ILP, the read operation and write operation must access at least one common replica. Moreover, any two write operations must access at least one common replica. Thus, we define the read (write) quorum to be the nodes of the R_type (W_type) loop in the intersect graph on the torus structure. Sites in distributed systems are assigned to nodes in the intersect graph according to the power of each node and the physical network connection.

Read locks and write locks are used in ILP. Also, we utilize some strategies such as two-phase locking, timestamp ordering and optimistic concurrency control to ensure one-copy serializability. Besides, the version number can help to show the order for data update. By the gathered data copies in different sites, the newest version of the object in the system is obtained.

At the beginning, the protocol issues a STATUS_CHECK message to all nodes in the intersect graph. When a node receives a STATUS_CHECK message, it must reply an acknowledgement associated with its present status. If a node does not respond during the expected time, then it is assumed damaged.

In a small system, every operation will send a STATUS_CHECK message to all the nodes of this system. And each node will respond its status. However, if we hope to avoid the waste of resources due to STATUS_CHECK in a large system, we can use the one-by-one inquiring or multi-level inquiring. We first check the node which is the easiest to form the quorum. If the quorum can not be formed since some nodes fail then we inquire the next node.

The response messages in the protocol are READ_LOCKED, WRITE_LOCKED and FREE. There are two cases which will generate an ABORT message: if the combination of nodes of READ_LOCKED and FREE can not form an R_type loop in the intersect graph for a read operation, and if the combination of nodes of FREE can not form a W_type loop in the intersect graph for a write operation.

The detailed replica control algorithm of ILP for getting the read and write quorum are listed in [6].

## 4. Analysis of The ILP Protocol

Nodes and links are not organized into a specific architecture in ILP. Similar to the coterie, we are not able to analyze the ILP itself, but analyzing some examples of ILP is possible. In this section, we will present the properties of ILP, reformulate several previous protocols by ILP.

### 4.1 Properties of ILP

In the intersect graph of ILP, since a read quorum is the set of nodes which belong to the H_type or V_type loops, if a node has many links of one type to get a loop of small length, then the node can easily connect with other nodes to get a read quorum. On the other hand, a write quorum is the nodes of the combination of any two different types of loops. If a node has many links of one type but has a few links of other types, then it is difficult to get a write quorum for the node. If the minimal quorum size is smaller, the efficiency of the system will be better.

### 4.2 Reformulation of Previous Protocols

Some proposed protocols can be reformulated by ILP. In the figures of this section, we add some virtual

nodes to present 3D torus structures on a plane paper. They just show the connections between the nodes on the boundaries. They are presented by dotted circles with lowercase characters.

#### 4.2.1 ROWA protocol:

In ROWA protocol, assuming the total number of nodes is N, read quorum is any one data copy, and write quorum is all data copies. Thus, ROWA protocol can be reformulated into Figure 4.1. We can observe that the minimal V_type and H_type loop sizes of each node are 1 and N, respectively. So any read request at any node just needs to lock the node and any write request at any node must lock all nodes. The number of V-type links of each node is 2N, and the number of H-type links of each node is 2, that means ROWA has high read availability and low write availability. The minimal read quorum size and minimal write quorum size are 1 and N, respectively.

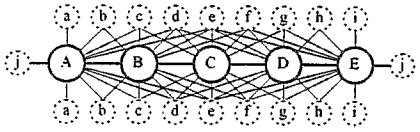


**Figure 4.1** An Example of ROWA Protocol Reformulated by ILP

#### 4.2.2 Grid protocol

In the grid protocol, the read quorum is a set of nodes that contains one node from each column or all nodes from one column of the grid. So we make H_type links connect each node to all the nodes on its adjacent columns, and each node on the leftmost column connects all the nodes on the rightmost column. Any V_type loop also contains all nodes on one column, so the V_type links connect the nodes on its adjacent rows and the top node on each column connects the bottom node on the column.

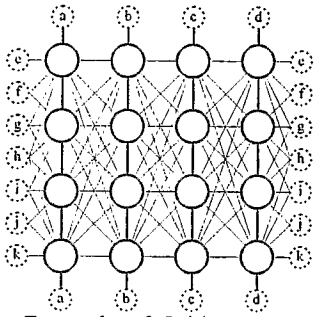

**Figure 4.2** An Example of Grid Protocol Reformulated by ILP

The grid protocol reformulated by ILP is shown in Figure 4.2. Assume that the dimension of the grid is m×n, and m≤n. The number of V-type links of each node is 2, and the number of H-type links of each node is 2m, that means the grid protocol has high read availability and low write availability. The minimal read quorum size and minimal write quorum size are m and m+n-1, respectively.

#### 4.2.3 Triangular lattice protocol

In the triangular lattice protocol (TLP), the read and write quorums use the idea of two kinds of path in which one connects the left and right boundaries and the other connects the top and bottom boundaries. A path connecting two boundaries in the original graph of TLP is the same as a loop containing this path. In the reformulated graph, the left and right boundaries or top and bottom boundaries are connected as the complete bipartite graph does.

An example of the triangular lattice protocol reformulated by ILP is shown in Figure 4.3. Nodes on boundaries have different responsibility from the nodes inside for controlling data access. Assume that the dimension of the grid is m×n, and m≤n. The upper-leftmost node and lower-rightmost node have most of the minimal R_type and W_type loops (their lengths are m and n+1, respectively), so their loads are the highest. The upper-rightmost node and lower-leftmost node have the least number of the minimal R_type and W_type loops (their lengths are m and m+n, respectively), so their loads are the lowest.

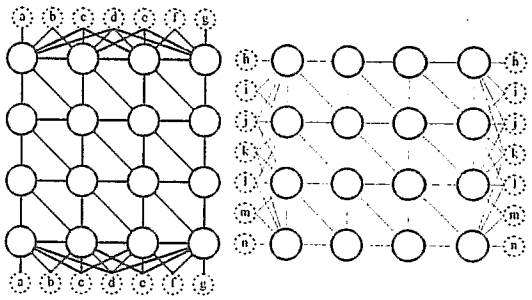

**Figure 4.3** An Example of Triangular Lattice Protocol Reformulated by ILP

## 5. Protocol Comparison Based on ILP

By ILP, we have a uniform way to analyze several proposed protocols. We write programs to test some proposed protocols reformulated by our ILP. Analysis results will be illustrated in this section.

#### 5.1 Test Samples

We write programs to test a 16 nodes(4x4) sample for the grid architecture(GA) and triangular lattice architecture(TLA) reformulated by ILP. We compute all properties of this 4x4 sample by enumerating all the possible cases for data copies to be available or not available. We will show the comparison results of the

two architectures of ILP with 16 nodes. The node number of this sample is shown in Figure 5.1. The detailed computational results of other architectures, such as 3x3 or 3x4 nodes, etc., are illustrated in [6].

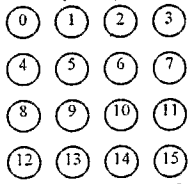


**Figure 5.1** Node Numbers of Test Samples

### 5.2 Results

We will analyze the following properties:

- *the read/write availability of a system*: the probability of the read(write) operation which can be successfully performed in a system.
- *the read(write) availability of a node*: the probability of the node which can perform the read(write) request.
- *the average read quorum size of a system*: the average read quorum size of every node in a system.
- *the average write quorum size of a system*: the average write quorum size of every node in a system.
- *the read(write) load rate of a node*: the ratio of read(write) load of this node to the minimal read(write) load of the node in the system.

#### 5.2.1 Read/Write Availability

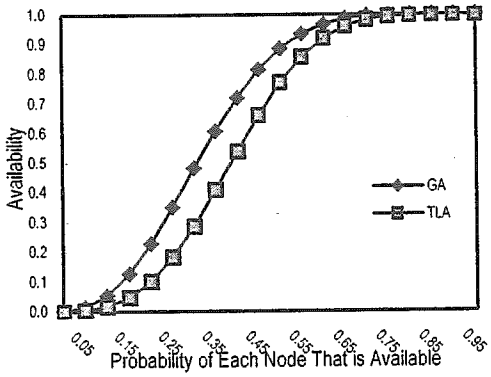


**Figure 5.2** Comparison of Read Availablity

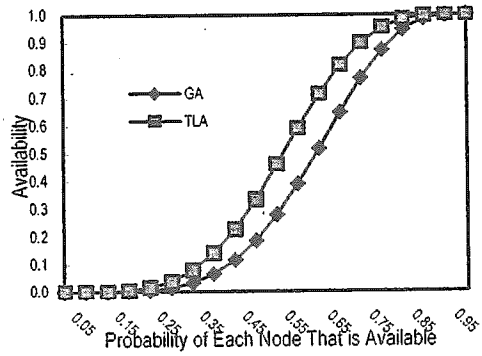


**Figure 5.3** Comparison of Write Availability

Figure 5.2 shows the read availability of the two protocols under different probability of each node that is available. We observe that GA is better than TLA. Figure 5.3 shows the write availability of the two protocols under different probability of each node that is available.

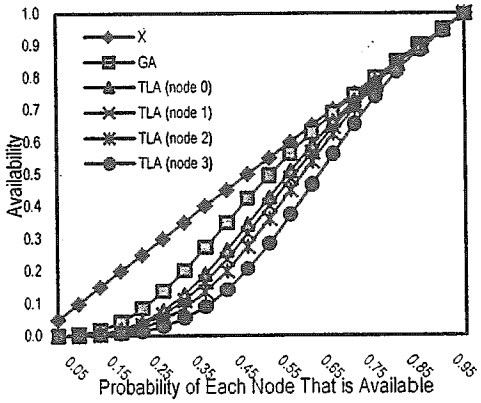


**Figure 5.4** Comparison of Read Availability of Each Node

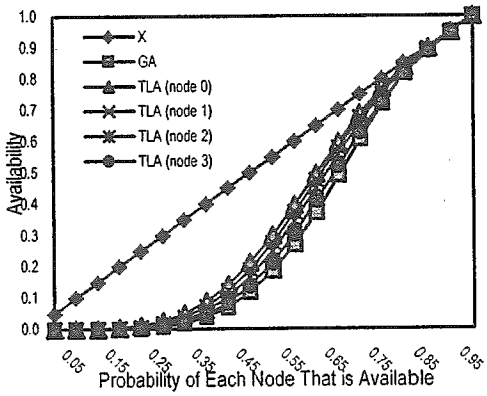


**Figure 5.5** Comparison of Write Availability of Each Node

Figure 5.4 shows the read availability of the two protocols under different probability of each node that is available. From this figure, we observe that GA is the best and the node 3 of TLA is the worst. Figure 5.5 shows the write availability of the two protocols under different probability of each node that is available. GA is better than the node 3 of TLA if the probability of each available node is higher than 0.85.

#### 5.2.2 Average Read(Write) Quorum Size

Figure 5.6 shows the average quorum size of the two protocols under different probability of each node that is available. We observe that the value of GA is fixed and the best. Figure 5.7 shows the average write quorum size of the two protocols under different probability of each node that is available. From this figure, we observe that TLA is better in general.

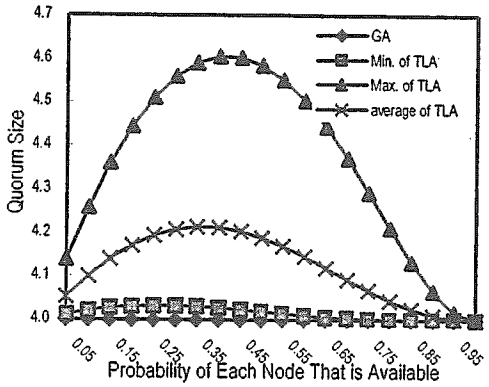


**Figure 5.6** Comparison of Average Read Quorum Size of A System and Some Nodes

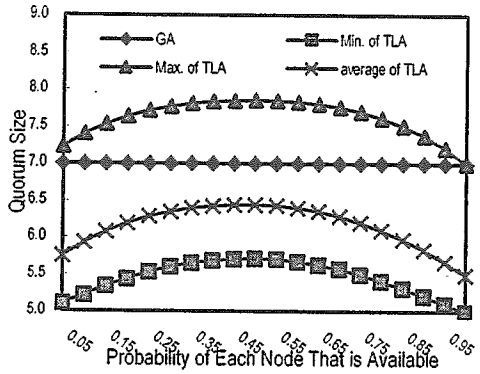


**Figure 5.7** Comparison of Average Write Quorum Size of A System and Some Nodes

### 5.2.3 Load Share Rate

Figure 5.8 shows the read load rate of some nodes of TLA under different probability of each node that is available. From this figure, we observe that the variance value of each node is small if the available probability of each node is high enough. Figure 5.9 shows the write load rate of some nodes of TLA under different probability of each node that is available. The variance value of each node is large, and the write load of node 0 is much larger than that of the other nodes.

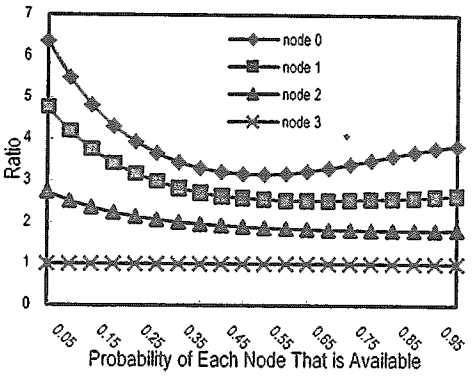


**Figure 5.8** Read Load Rate of Some Nodes of TLA

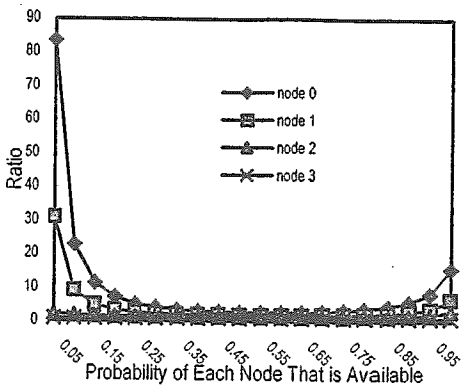


**Figure 5.9** Write Load Rate of Some Nodes of TLA

## 6. Conclusions and Future Work

In this paper, we propose a new protocol named intersect loops protocol (ILP). Many proposed protocols, such as the ROWA protocol, grid protocol, and triangular lattice protocol, can be reformulated by ILP. The future work includes analyses for broder types of other proposed protocols which use the concept that the quorum must contain the majority of something in the system.

## References

[1] D. Agrawal and A. El Abbadi, Exploiting Logical Structures in Replicated Databases, *Information Processing Letter*, Number 33, pages 255-260, 1990.

[2] P.A. Bernstein and N. Goodman, An algorithm for Concurrency Control and Recovery in Replicated Distributed Database, *ACM Transactions on Database Systems*, Volume 9, Number 4, pages 596-615,1984.

[3] P.A. Bernstein, V. Hadzilacos and N. Goodman, *Concurrency Control and Recovery in Database System*, Addison-Wesley, Reading, MA, 1987.

[4] G. Coulouris , J. Dollimore and T. Kindberg, *Distributed Systems Concepts and Design*, 2nd edition, Addison-Wesley, 1994.

[5] A. Kumar, M. Rabinovich and R. Sinha, A Performance Study of New Grid Protocol and General Grid Structures for Replicated Data, Tech. Report 93-03-02, Department of Computer Science and Engineering, University of Washington, March 1993.

[6] W.Y. Wang, Intersect Loops Protocol: A General Protocol for Replica Control in Distributed System, Master Thesis, National Chung-Hsing University, Taiwan, 1997.

[7] C. Wu and G.G. Belford, The Triangular Lattice Protocol: A Highly Fault Tolerance and Highly Efficient Protocol for Replicated Data, *Proceeding of the IEEE 11th Symposium on Reliable Distributed Systems*, pages 66-73, 1992.

[8] C. Wu and G.G. Belford, Replica Control Protocols that Guarantee High Availability and Low Access Cost, Technical Report UIUCDCS-R-93-1817, Universal of Illinois, July 1993