Order Number 9401667

The recovery of locality for causal sets and related topics

Daughton, Alan Ronald, Ph.D.

Syracuse University, 1993

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

The theory of causal sets is an attempt at a successful quantum theory of gravity. It is widely expected that any theory of quantum gravity should give rise to a discrete structure to spacetime geometry. This supposition rests upon three features of successfully quantized theories. The first is that quantization usually leads to some form of discrete object, or quantum, which is the source of the term "quantization". The second is the need for renormalization of field theories (which is not alleviated by including gravity, using the usual perturbative approach to quantization, as had once been hoped). An ultraviolet cutoff is introduced, which can be imagined to be either the mathematical manifestation of an underlying theory that contains new structures at some scale (for example the "GUT" fields, strings, etc.) or simply the result of the breakdown of the manifold picture at this same scale. Thirdly, quantized fields exhibit the phenomenon of virtual particles of unlimited energies at short length scales. The coupling of gravity to such quantized fields should induce infinite curvatures of the manifold at short scales, thereby dynamically prohibiting a manifold structure. The theory of causal sets begins with a discrete structure. The classical spacetime manifold is expected to emerge as an approximation (in a large grain limit) to the class of causal sets which dominate a sum over causal sets, each causal set being weighted by the exponentiation of an appropriately defined "action functional".

In order for this theory to be successful, several questions need to be addressed. Some of these are directed at the relationship between the causal set theory and the geometry of classical spacetime. Other questions are related to the quantization of the causal set theory and inclusion of quantum fields into the theory's framework. This work addresses the relation between causal sets and classical gravity (specifically the issues of embeddability of a certain class of causal sets known as the binomial partial orders into manifolds), and the recovery of locality for those causal sets which are embeddable into classical manifolds, a necessity for the successful behavior of the theory in the classical regime as well as the inclusion of quantum fields.

This work is divided into five main parts. The first gives an overview of the theory of causal sets. The second deals with the issue of the embeddability of the binomial partial order into manifolds. The third part discusses the locality issue. The fourth treats the still

unsolved problem of enumerating causal sets consisting of a finite number of elements. The fifth contains a discussion of some open questions and speculations on the further progress of the theory. The work finishes with two appendices listing the computer codes used.

THE RECOVERY OF LOCALITY FOR CAUSAL SETS AND RELATED TOPICS

by

Alan Ronald Daughton

B.S., California Institute of Technology,

Pasadena, CA, 1986

M.S., Syracuse University, 1989

DISSERTATION

Submitted in partial fulfillment of the requirements

for the degree of Doctor of Philosophy in Physics

in the Graduate School of Syracuse University

May 1993

Approved Rafael Sorkin

Professor Rafael Sorkin

Date 26 April 1993

The Graduate School
Syracuse University

We, the members of the Oral Examination Committee, hereby approve the thesis/dissertation of

Alan Daughton
Candidate's Name

whose oral examination was held on

April 16, 1993
Date

Aiyalam Balachandran
Examiner *(Please type)*

Aiyalam Balachandran
(Please sign)

Joshua Goldberg
Examiner *(Please type)*

Joshua Goldberg
(Please sign)

Peter Saulson
Examiner *(Please type)*

Peter R. Saulson
(Please sign)

Joseph Schechter
Examiner *(Please type)*

Joseph Schechter
(Please sign)

Rafael Sorkin
Thesis/Dissertation Advisor
(Please type)

Rafael Sorkin
(Please sign)

Jack Graver
Chair, *(Please type)*
Oral Examination Committee

Jack E Graver
(Please sign)

ACKNOWLEDGEMENTS

I would like to thank Rafael Sorkin for the inspiration and encouragement he has given me. He is a fountain of interesting ideas, and I am continually amazed at the ease with which he can fit them together in such a beautifully coherent system. I feel privileged to be working with him on a small fraction of these ideas. His patience and understanding have been helpful to me as I've struggled (and still struggle) to grasp all that he has to say.

I would also like to thank David Meyer and Eric Woolgar for many enlightening discussions. I don't believe either are aware of the great amount that I've learned from them.

Finally, I owe a special debt of gratitude to my parents who got me started on the right path in life, and have been a great support over the *many* years of my education.

TABLE OF CONTENTS

Part I

Introduction to Causal Sets

The theory of causal sets is based upon the mathematics of partial orders. A set S has a partial order defined on it if there is a relation $\prec$ between elements of S that obeys the following rules:

$$\text{irreflexivity}: \quad x \not\prec x,$$

$$\text{transitivity}: \quad x \prec y \quad \text{and} \quad y \prec z \Longrightarrow x \prec z.$$

[A more standard definition has reflexivity, $x \prec x$, with the added condition that if $x \prec y$ and $y \prec x$ then $x = y$.][1)]

When the set S is discrete the partial order can be pictured as a graph. Some graphs of this type will be presented in this work, using the convention that $a \prec b$ will be diagrammed so that a will be higher on the page than b and there will be a path from a to b which is always upward directed.

As a simple but non-trivial example, consider the set $S = \{a, b, c, d, e, f\}$ with the partial order:

$$a \prec \{b, c, d, e, f\},$$
$$b \prec \{d, e\},$$
$$c \prec \{d, e, f\},$$
$$d \prec \{e\}.$$

This is depicted in Figure 1.

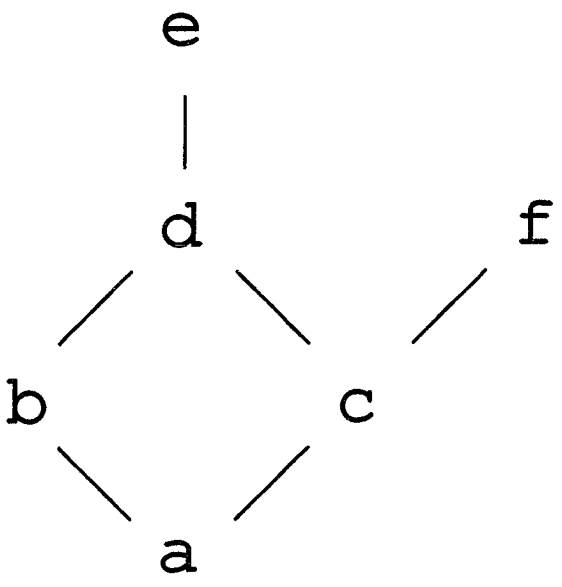

Figure 1

Notice that only those relations which are not implied by transitivity have been drawn; such relations will be called links. Also note that while f appears higher on the page than b, they are not related since there is no path from b to f which is always upward directed.

In calculations, partially ordered sets are represented by matrices. One assigns a labelling to the points by integers and then defines the causal (aka incidence) and link matrices. The causal (incidence) matrix C has element

$$\mathrm{C}_{ij} = \begin{cases} 1, & i \prec j \\ 0, & \text{otherwise.} \end{cases}$$

The link matrix L has element

$$\mathrm{L}_{ij} = \begin{cases} 1, & i \prec j \text{ and } \nexists k \text{ s.t. } i \prec k \prec j \\ 0, & \text{otherwise.} \end{cases}$$

It is easily demonstrated that

$$L = C - C^2,$$

where matrix multiplication is done using Boolean arithmetic. [Recall that Boolean arithmetic has the numbers 0,1 replaced by the logical values false,true and the operations of multiplication and addition replaced by the operations of AND and OR.]

For the partially ordered set in Figure 1, C and L are:

$$C = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}, L = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

with the labelling

$$\begin{array}{ll} a \leftrightarrow 1 & d \leftrightarrow 4 \\ b \leftrightarrow 2 & e \leftrightarrow 5 \\ c \leftrightarrow 3 & f \leftrightarrow 6. \end{array}$$

Before introducing causal sets it will be useful to define the concept of local finiteness. A partially ordered set S is locally finite if for every a , $c \in$ S, there are a finite number of elements $b \in$ S such that $a \prec b \prec c$
(b is said to lie in the interval $[a, c]$). Notice that any locally finite partially ordered set is necessarily discrete.

A simple example is S=Z, the set of integers with $m \prec n$ iff $m < n$. This is locally finite as the number of elements j in Z such that $i \prec j \prec k$
for $i, k \in$ Z is $k - i - 1$, a finite number. This is an example of a locally finite partially ordered set with an infinite number of elements.

A causal set is a locally finite partially ordered set. Causal sets are physically motivated by the observation that for Lorentzian manifolds (i.e. manifolds upon which an everywhere Lorentzian metric is defined), the relation between two points of being causally related satisfies the properties of a partial order. This gives a partial order to any Lorentzian manifold.

[Actually, partial orders do not admit structures which are discrete analogs of closed time-like curves. This is because any two points, say x and y, on such a curve will have both $x \prec y$ and $y \prec x$. In the first definition of a partial order relation these relations and transitivity imply $x \prec x$ (and $y \prec y$) which violates the condition of irreflexivity. The second definition of a partial order relation demands that these two relations imply that $x=y$, and so the entire closed timelike curve will be represented by a single point in the partial order. Thus, any Lorentzian manifold which contains closed timelike curves cannot be a large grain approximation to a causal set. This is either an advantage or disadvantage depending on one's taste for closed time-like curves.]

Since any manifold with a well behaved (e.g. globally hyperbolic) metric will have a finite volume Alexandrov neighborhood between causally related points which are sufficiently "close" to each other (in a topological sense), this allows a connection between the notions of volume of an Alexandrov neighborhood and number of points in an interval. The specifics of this connection will be discussed below.

A causal set S is embeddable into a Lorentzian manifold M if there is a one to one mapping $\phi : S \mapsto M$ between the elements of the causal set and a discrete set of points in the manifold such that when $a \prec b$, $a, b \in S$, then $\phi(b) \in \overline{J^+(\phi(a))}$, where $\overline{J^+(p)}$ is the set of points in M that are to the causal future of $p \in M$.

Given a manifold M with a volume element dV, a sprinkling is a random set of points of M chosen according to a Poisson process, i.e. the expected number of points in any region R is $= \int_R dV$.

A causal set S is faithfully embeddable into a Lorentzian manifold M if there is a one to one mapping $\phi^{(F)} : S \mapsto M$ such that $\phi^{(F)}(S)$ is a sprinkling of M.

[The theory of causal sets postulates the causal set as the primary physical object. The manifolds of classical physics are imagined to emerge in some appropriate limit of a quantum theory of causal sets. The interpretation of partial order relations in the causal set going over to causal relations of a manifold provides a natural explanation for the observed signature of the metric. Causal sets will only be faithfully embeddable into manifolds with one time-like direction. Nothing can be said about the dimension of the manifold (i.e. the number of space-like directions). This information will have to emerge dynamically.]

Given a partially ordered set S, a subposet $S' \subset S$ is a partially ordered set (given as a subset of S) with the order relations inherited from those of S.

One method of coarse-graining a causal set S is to select a subposet S'. The selection can be arbitrary or according to some set of rules. Other methods of coarse-graining can be imagined.

The theory of causal sets, when complete, will include some prescribed method of coarse-graining so that the sum over causal sets will be dominated by causal sets which have coarse-grainings that can be faithfully embedded into a Lorentzian manifold (at a reduced density corresponding to the degree of coarse-graining).

To illustrate this vaguely expressed notion, imagine an action functional which is extremized by an infinite chain of diamond-shaped link structures, as illustrated in Figure 2.

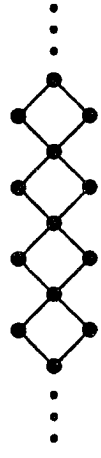

Figure 2

This causal set can be embedded in 2-dimensional Minkowski spacetime M^2, but not faithfully so. This is because there is no sprinkling of M^2 which looks like a string of diamonds. If we choose a subposet S' to consist of, say, all the points where the diamonds touch then S' can be faithfully embedded into 1-dimensional Minkowski spacetime M^1 (which is just R^1 with metric $ds^2 = -dt^2$).

This concludes the discussion on the general ideas of the causal set theory. The re-

maining parts of this work will focus on more specific details of the program.

Part II

The Embeddability of the Binomial Poset

Given a causal set, one would like to ascertain a number of its features. Among these is the issue of whether the causal set can be embedded into a manifold and the topological properties of any manifolds (if any) into which the set can be embedded. An important feature of a manifold's topology is its dimension. One might make a determination of this by locating subposets of the causal set to be embedded which are "markers" of dimension, i.e. subposets which can only be embedded into manifolds of some minimum dimension. This part explores the embedding question for a class of posets known as the binomial posets ("binomial" as they consist of 2^N elements arranged in layers, the M^{th} layer containing $\binom{N}{M}$ elements, for integers M, N) into some simple (i.e. containing some symmetries) manifolds. The binomial posets are also interesting because *any* poset can be viewed as a subposet of *some* binomial poset.

The binomial poset on N elements, B_N, consists of all subsets of the N elements, with the partial order relation being that of inclusion.

The binomial set on 1 element, B_1, consists of all the subsets of $\{a\}$.
So $B_1 = \{\emptyset, a\}$ with $\emptyset \prec a$.

B_2 consists of all subsets of $\{a, b\}$. So $B_2 = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$ with

$$\emptyset \prec \{\{a\}, \{b\}, \{a, b\}\},$$
$$\{a\} \prec \{a, b\},$$
$$\{b\} \prec \{a, b\}.$$

B_3 consists of all subsets of $\{a, b, c\}$. So $B_3 = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$ with

$$\emptyset \prec \{\{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\},$$
$$\{a\} \prec \{\{a, b\}, \{a, c\}, \{a, b, c\}\},$$
$$\{b\} \prec \{\{a, b\}, \{b, c\}, \{a, b, c\}\},$$

$$\{c\} \prec \{\{a,c\},\{b,c\},\{a,b,c\}\},$$
$$\{a,b\} \prec \{a,b,c\},$$
$$\{a,c\} \prec \{a,b,c\},$$
$$\{b,c\} \prec \{a,b,c\}.$$

The graphs for these are shown in Figure 3.

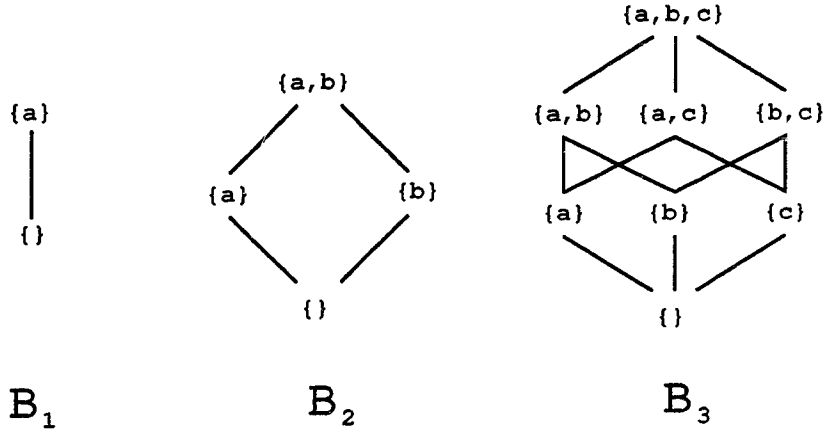

Figure 3

Higher B_N's are constucted analogously.

An alternative way of defining the binomial poset B_N is possible. Consider an orthonormal set of vectors $\{\hat{x_i}\}$ in R^N, where $1 \leq i \leq$N. Then each element of B_N corresponds to a vector sum

$$\vec{v}^{(k)} = \sum_{i=1}^{N} a_i{}^{(k)} \hat{x_i},$$

where the $a_i{}^{(k)} \in \{0,1\}$. The order relation is

$$\vec{v}^{(k)} \prec \vec{v}^{(l)} \text{ if } \quad a_i{}^{(k)} \leq a_i{}^{(l)} \quad \text{for all} \quad 1 \leq i \leq N.$$

Thus, B_N consists of 2^N elements and the $a_i{}^{(k)}$ provide a map from the binary encoded

integers in the interval [0,2^N-1] to B_N. Also, the 2^N $\vec{v}^{(k)}$'s have their tips at the vertices of an N-dimensional cube.

Mathematicians use such a vector construction to show that B_N is embeddable into R^N.[2] [The ordering of R^N is given by the rule that $x \prec y$ for $x,y \in R^N$ if each entry of the N-tuple of numbers that make up x is not greater than each entry of the N-tuple that makes up y.] The theory of causal sets interprets the partial order relations as causal relations. If we choose

$$\vec{v}^{2^N-1} = \sum_{i=1}^{N} \hat{x}_i$$

to be the future timelike direction of N-dimensional Minkowski space M^N, then B_N is embeddable into M^N using "square" lightcones, consisting of an N−1 simplex with linearly increasing scale as time increases. Physical lightcones are not square, but round, consisting of an N−2 sphere with linearly increasing radius as time increases. The question arises: Is B_N embeddable into M^N (or some other manifold) using the physically relevant round lightcones? (So that the order relations correspond to causal relations.)

If the answer is "yes", then locating binomial subposets in a causal set can yield information on the minimum dimension of a manifold into which the causal set can be embedded. If the answer is "no", then the binomial poset will provide a counterexample to the hypothesis that any causal set can be embedded into Minkowski space of *some* dimension.

It was demonstrated by Meyer[3] that the answer is "no" for N≥6 (into M^N, and if B_N does not embed into M^N it also does not embed into M^P for *any* P), when only symmetric embeddings are considered. This result also means that B_N cannot be symmetrically embedded into M^P for any $P \geq N$, as the points corresponding to the N vectors $\{\hat{e}_i\}$ defines an N dimensional timelike plane in M^P within which the embedding cannot be carried out, and the extra dimensions play no role for symmetric embeddings. The question of non-symmetric embeddings is still open. A simpler demonstration of the symmetric case is presented here.

A space+time decomposition of M^N is $M^N = R^{N-1} \times R$. To embed B^N into M^N, consider the embedding of the symmetric $N-1$ simplex into R^{N-1}. This embedding consists

of N vectors $\vec{e}_i, 1 \leq i \leq N$, such that

$$\vec{e}_i \cdot \vec{e}_j = N\delta_{ij} - 1,$$

where δ_{ij} is the Kroneker delta. Each element of B_N will have a spatial projection which is the barycenter of some sub-simplex of the $N-1$ simplex.

As a clarifying example, consider B_4. The 3-simplex in R^3 is given by 4 vectors such that $\vec{e}_i \cdot \vec{e}_j = 3$ when $i = j$ and -1 when $i \neq j$.

The point corresponding to the binary number 0000 is the origin. The points corresponding to 0001, 0010, 0100, 1000 are the vectors $\vec{e}_i$, with time coordinate $\sqrt{3}$ so that lightrays originating at the origin in M^4 will arrive (at time $\sqrt{3}$) at the four points $\vec{e}_i$ (which are the barycenters of the 0-dimensional subsimplices). Lightrays originating at these points will begin to overlap at $\frac{1}{2}(\vec{e}_i + \vec{e}_j)$ where $i \neq j$. The time coordinate will be assigned the appropriate value. These six points are the barycenters of the 1-dimensional subsimplices. Lightrays originating from these six points will have overlaping lightcones (in groups of 3) at the four barycenters of the 2-dimensional subsimplexes. Again, the time coordinates will be chosen appropriately. Lightrays emanating from these four points will converge at the center of the 3-dimensional simplex.

Now, does this embedding have the causal relations matching the partial order relations of B_4? The embedding was set up so that all points in B_4 that are related will be causally related in M^4. We must check that points in B_4 which are not related must be spacelike separated in M^4. Only one relation needs to be checked, that between the points $\vec{e}_1$ and $\frac{1}{3}(\vec{e}_2 + \vec{e}_3 + \vec{e}_4)$. This is the only one to be checked because any other "spurious" causal relation that might occur will enforce a causal relation between these points (and, of course, by the symmetry of the arrangement). If, then, these points are spacelike related then all causal relations will match the relations of the partial order B_4. These points will be spacelike related if the time coordinate of $\frac{1}{3}(\vec{e}_2 + \vec{e}_3 + \vec{e}_4)$ minus $\sqrt{3}$ is less than the spatial distance between $\vec{e}_1$ and $\frac{1}{3}(\vec{e}_2 + \vec{e}_3 + \vec{e}_4)$.

Physically, this amounts to a race between two lightrays which originate at $\vec{e}_1$, one travelling directly to $\frac{1}{3}(\vec{e}_2 + \vec{e}_3 + \vec{e}_4)$ and the other travelling to $\frac{1}{2}(\vec{e}_1 + \vec{e}_2)$ and then proceeding to $\frac{1}{3}(\vec{e}_1 + \vec{e}_2 + \vec{e}_3)$. If the direct path loses the race then the points are

spacelike separated and the causal relations match the relations of B_4.

The general case will follow after the introduction of some notation.

Let $p \subset \{1, \ldots, N\}$, and $|p|$ = cardinality of p.

Let

$$\vec{e}_i \cdot \vec{e}_j = N\delta_{ij} - 1$$

define N vectors $\vec{e}_i$, $1 \leq i \leq N$.

Define

$$\vec{v}_p \equiv \frac{1}{|p|} \sum_{i \in p} \vec{e}_i \, .$$

$\vec{v}_p$ is the barycenter of the $|p|-1$ simplex bounded by vertices p. A short calculation gives

$$\vec{v}_p \cdot \vec{v}_q = \frac{|p \cap q|N - |p||q|}{|p||q|}.$$

Let $p \subset q$ such that $|p| = |q| - 1$. Then the distance between points $\vec{v}_p$ and $\vec{v}_q$ is

$$\mathrm{d}_{pq} = \sqrt{\frac{N}{|p|(|p|+1)}}.$$

Let $|p| = 1, |q| = N - 1$, and $p \cap q = \emptyset$. Then

$$\mathrm{d}_{pq} = \frac{N}{\sqrt{N-1}}.$$

For B_N to be embeddable into M^N we must consider the race between two lightrays emanating from $\vec{e}_1$, one going directly to

$$\frac{1}{N-1} \sum_{i=2}^{N} \vec{e}_i$$

and the other travelling from $\vec{e}_1$ (a 0-simplex) to each of the higher simplexes (containing $\vec{e}_1$ as a vertex) in succession. If the ratio

$$R_N \equiv \frac{\sum_{l=1}^{N-2} \sqrt{\frac{N}{l(l+1)}}}{\frac{N}{\sqrt{N-1}}} < 1,$$

then B_N is embeddable into M^N. The values are:

$$R_3 = .577350269,$$
$$R_4 = .965925825,$$
$$R_5 = 1.255802792,$$
$$R_6 = 1.485822502.$$

Actually, for $|p| \geq 2$, the points in the embedding can be chosen to be

$$r_{|p|}\ \vec{v}_p\ , \text{for} \quad r_{|p|} > 0.$$

The embedding is still symmetric. With an appropriate choice of the scale factors $r_{|p|}$'s Meyer showed that B_5 can in fact be embedded, but there is no choice which allows a symmetric embedding of B_N for $N \geq 6$.

Perhaps B_N can be embedded into some curved space. This work gives a partial answer to this possibility.

Viewing the problem as a race, one can see that in M^N the direct path is purely radial, passing through the origin of R^{N-1}. The wandering path, on the contrary, is to a large extent tangential, with each component of the path having a larger tangential component than radial component. Thus one might expect that in positively curved spaces, where tangential distances are relatively shortened with respect to radial distances, higher B_N's might be embeddable.

Consider, then, the manifold $S^{N-1}\times R$ with spherical metric on S^{N-1} and with the radius parameter R $(= \sqrt{N-1})$ constant in time. For maximal effect, the points of B_N (except for points corresponding to binary 0 and 2^{N-1}) will be embedded into the equatorial plane of S^{N-1}, i.e. into S^{N-2}. If S^{N-2} is considered as a surface in R^{N-1}, then once again we can define N vectors $\vec{e}_i$ such that

$$\vec{e}_i \cdot \vec{e}_j = N\delta_{ij} - 1,$$

and $2^N - 1$ vectors $\vec{v}_p$ by

$$\vec{v}_p \equiv \frac{N-1}{|p|(N-|p|)} \sum_{i \in p} \vec{e}_i \, .$$

Now distances are arclengths, as tangential lightrays are constrained to lie on the surface S^{N-2} and the radial lightray will also lie on a different surface S^{N-2}, but of the same radius.

Let $p \subset q$ such that $|q| = |p| + 1$. Then the distance between points $\vec{v}_p$ and $\vec{v}_q$ is

$$d_{pq} = R \arccos \frac{\vec{v}_p \cdot \vec{v}_q}{|\vec{v}_p||\vec{v}_q|}$$

$$= R \arccos \sqrt{\frac{|p|(N-(|p|+1))}{(N-|p|)(|p|+1)}}.$$

Let $|p| = 1, |q| = N-1$, and $p \cap q = \emptyset$. Then

$$d_{pq} = \pi R.$$

Now consider the ratio

$$R_N \equiv \frac{1}{\pi} \sum_{l=1}^{N-2} \arccos \sqrt{\frac{l(N-(l+1))}{(N-l)(l+1)}}.$$

If $R_N < 1$ then B_N is embeddable into $S^{N-1} \times R$. The values are:

$$R_3 = \tfrac{1}{3},$$
$$R_4 = .608173448,$$
$$R_5 = .848151096,$$
$$R_6 = .977960384,$$
$$R_7 = 1.262117426.$$

Thus, B_6 is embeddable but B_N for N$\geq$ 7 is not. Now the trick of considering the embedding with scale factors $r_{|p|}$ is less likely to work for B_7 as this process involves using lightpaths which are less tangential and the gain achieved from curvature is weakened. Anyway, it's clear that there is some N for which no embedding can be done.

One might consider keeping the radius of the equatorial plane, R, constant while increasing the "radius" of the radial direction (i.e. making S^{N-1} prolate), but the path between $\vec{e}_1$ and $\frac{1}{N-1}\sum_{i=2}^{N} \vec{e}_i$ which lies in the equatorial plane still remains of constant length πR.

Also, one might consider R=R(t), as for example the de-Sitter spacetime. This will have no effect upon whether $R_N > 1$ or $R_N < 1$, however. R_N is a ratio of two distances along two different paths. Each distance is an integral along the respective path of the line element of the induced spatial metric. At any given time t, the line element is the same along both paths as it is scaled by the global factor R(t), and hence the integrals are the same to the time when the shorter path has been completed. At this time, the shorter of the two distances ceases to increase, becoming a constant for all subsequent times. The larger of the two distances continues to increase until the time that the longer path is completely traversed. At that time the ratio of the two lengths is the number R_N. Thus, introducing the factor R(t) cannot change the shorter to the longer path and vice versa, and so the ratio R_N will be greater or lesser than one regardless of R(t).

Thus, no symmetric embeddings of B_N, for sufficiently large N, into $S^{N-1}\times$R exists. Again, the question of non-symmetric embeddings remains open.

The embeddability of B_N was investigated in an attempt to find an easily identifiable indicator of the dimension of causal sets which may be embeddable into some manifold. The search for embeddings into M^N was motivated by the notion that embeddings will be sufficiently dense that binomial subposets of a causal set will be embedded into regions of small enough volume that the manifold is approximately flat. Since B_6 cannot be embedded into M^6, its presence as a subposet precludes the embedding of that portion of a causal set into a small flat region. B_6 can be embedded into a large portion of $S^{N-1}\times$R, however, and so its presence as a subposet could indicate a localized high curvature in a manifold into which the poset is to be embedded. As higher N binomial subposets cannot

be so embedded, what their presence forshadows for any manifold in which they are to be embedded is still unclear.

Another obvious spacetime into which one might attempt to embed B_N is the Schwarzschild solution. This has not yet been done. Should this fail as well, one might strongly suspect that there is some N beyond which no B_N can be embedded into any manifold.

Should this prove to be true, then this fact might be useful in selecting an appropriate action functional, as a proper action should then be dominated by causal sets which have the property that some coarse-graining technique will eliminate large N B_N subposets, resulting in a set which should be embeddable into some manifold.

However, there is a heuristic reason to suspect that B_N can be embedded into a Schwarzschild solution. The tangential lightrays will all be exterior to the event horizon by a finite amount as long as the first N points are exterior to the horizon. However, the "radial" lightray will tend to graze the horizon. Thus the radial path will experience a time-delay with respect to the Schwarzschild coordinates. One can make this delay arbitrarily large by bringing the first N points closer to the horizon. The tangential paths will also experience a gravitational redshift, but to a lesser degree. This may be a mechanism which makes the embedding work.

Such a situation is somewhat reminiscent of some of the ideas of quantum gravity that suppose that at the Planck length the spacetime manifold breaks up into a foam of blackholes. As the causal set action principle will involve a coherent sum of all those causal sets which upon coarse-graining embed into the same manifold, those causal sets which have all binomial posets as subposets replaced by links in the coarse-graining will contribute to the "stationary point" amplitude. Since in manifold terms these B_N's might be considered to be micro-scale blackholes, the foam picture emerges naturally.

The above picture implies that links are "local" objects in an appropriate sense. Part III will argue that links cannot be so viewed. However, that discussion is limited to flat space, where Lorentz invariance coupled with discreteness makes locality an elusive concept. For curved manifolds, however, it is possible to define locality in terms of regions over which the curvature varies slowly. It is subposets embedded into such regions that upon coarse-graining over (e.g. replaced by a link) may be considered micro-scale objects.

A coarse-graining rule and dynamical action which combine to have the effect of only coarse-graining over such micro-scale structures would be desirable. One consequence of such might be that fields cannot be defined on flat space (if the ideas of Part V should prove to have merit) in accordance with Einstein's equation, as in this case no links can be determined to be local via the above criterion and so no structures (which are made of links) will be deemed micro-scale and hence no structures will be coarse-grained over.

Part III

Locality

The goal of the causal set theory is to provide a satisfactory quantum theory of gravity. Any such theory must contain the classical theory as an appropriate limit. Specifically, the classical limit should be characterized by a manifold upon which matter (including gauge) fields are defined in such a way that an Einstein-like equation is satisfied. Such an equation has the form

$$\delta \int dV \Phi(q(p), \partial q(p), \ldots) = 0,$$

where $q(p)$ is the collection of fields (including the metric) defined at each point p of the manifold, and $\partial q(p)$ are the derivatives of the $q(p)$'s. The "..." is meant to represent the possibility of higher derivatives entering the action.

The above equation is local, in that it involves a single volume integral of a one-point function. This form has the property that if $q(p)$ is replaced by $q(p) + \Delta q(p)$, where

$$\Delta q(p) = \Delta_1 q(p) + \Delta_2 q(p)$$

such that

$$\Delta_1 q(p) = 0 \quad \text{if} \quad p \notin R_1,$$

and

$$\Delta_2 q(p) = 0 \quad \text{if} \quad p \notin R_2$$

for disjoint regions R_1, R_2 (i.e. $R_1 \cap R_2 = \emptyset$), then

$$\int dV \Phi(q(p) + \Delta q(p), \partial[q(p) + \Delta q(p)], \ldots)$$

$$= \int dV \Phi(q(p), \partial q(p), \ldots)$$

$$+\int_{R_1} dV[\Phi(q(p)+\Delta_1 q(p),\partial[q(p)+\Delta_1 q(p)])-\Phi(q(p),\partial q(p))]$$

$$+\int_{R_2} dV[\Phi(q(p)+\Delta_2 q(p),\partial[q(p)+\Delta_2 q(p)])-\Phi(q(p),\partial q(p))].$$

Thus, the resulting change in the integral is the sum of the changes due to each of the regions R_1 and R_2 seperately.

In addition to locality, the classical equation is such that $\Phi(q(p),\partial q(p),\ldots)$ is Lorentz invariant. One could consider Φ's which are not Lorentz invariant, but they would still be local. Alternatively, one could consider other forms of the equations which are Lorentz invariant but not local, for example

$$\delta\int dV_1\int dV_2\Psi(q(p_1),\partial q(p_1),q(p_2),\partial q(p_2),\tau(p_1,p_2)),\ldots)=0$$

where $\tau(p_1,p_2))$ is some "field" which depends upon (say) the proper distance between points p_1 and p_2.

Both properties of locality and Lorentz invariance are imposed by experiment. Any adequately defined theory of quantum gravity will have both of these features in the classical sector.

There is a tension between these properties when discrete structures (instead of manifolds) are used as a basis for a theory. An illustrative example is that of a square lattice in $1+1$ dimensional Minkowski spacetime. On such a lattice (as for example in lattice gauge field theories) locality is achieved by having only near neighbor interactions. In the continuum limit (lattice spacing going to zero) such interactions will become local. This lattice breaks Lorentz invariance in an obvious way, as it defines a preferred frame, that for which the lattice is made up of squares.

In fact, any regular lattice will break Lorentz invariance in a similar way. The theory of causal sets has as one of its tenets the notion of faithful embeddings (in the classical sector at a minimum). As faithful embeddings are onto sprinklings of a manifold, and sprinklings are Lorentz invariant (the volume of a region is a Lorentz invariant object), the theory of causal sets has Lorentz invariance built in. But then there is a tension with locality, as points which appear to be near neighbors in one frame will be boosted to be arbitrarily distant in other frames (although, of coarse, their proper distance will

be unaffected). A consequence of this is that in any frame there are a large number of long, near null links. Figure 4 shows a causal set derived by a sprinkling of 50 points from 1+1 Minkowski spacetime, with link relations drawn. This dramatically illustrates that the majority of links appear highly non-local. As long as the continuum limit is not taken, this tension will persist no matter how dense the sprinkling. As the theory of causal sets specifically does not allow the taking of the continuum limit the locality issue remains to be dealt with. (In continuum theories it is the differential structure of manifolds which allow the peaceful coexistence of locality and Lorentz invariance.)

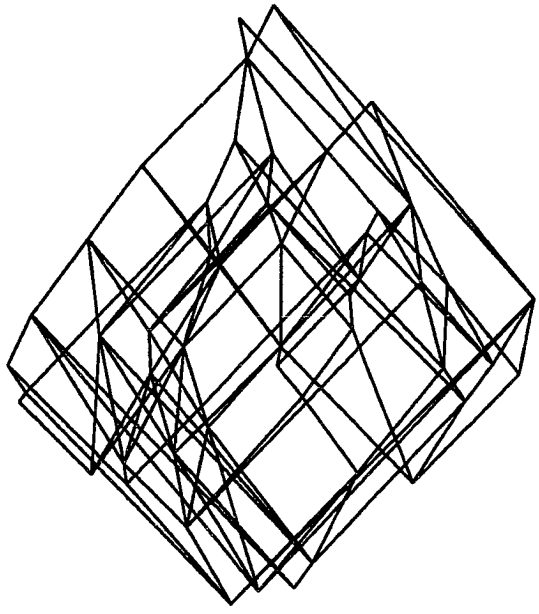

Figure 4

All may not be lost, however. Perhaps locality can emerge via a route different from that of "near neighbors". One possible route is explored in this work.

Given that locality is recoverable in the causal set framework, it has been shown[4] by dimensional arguments that any local action on causal sets must reproduce classical gravity in the appropriate limit. This in itself is a strong motivation for examining the locality issue.

Consider a massless scalar field $\phi(x)$ defined on 1+1 dimensional Minkowski spacetime. The action for the field is

$$\begin{aligned} S &= -\frac{1}{2}\int dV g^{ab}\nabla_a\phi(x)\nabla_b\phi(x) \\ &= -\frac{1}{2}\int dV \nabla_a[g^{ab}\phi(x)\nabla_b\phi(x)] \\ &\quad +\frac{1}{2}\int dV g^{ab}\phi(x)\nabla_a\nabla_b\phi(x) \\ &= -\frac{1}{2}\int_{\partial V} dS_a\phi(x)\nabla^a\phi(x) \\ &\quad +\frac{1}{2}\int dV\phi(x)\Box\phi(x) \end{aligned}$$

where $\Box \equiv g^{ab}\nabla_a\nabla_b$.

If $\phi(x) = 0$ and/or $\nabla^a\phi(x) = 0$ when $x \in \partial V$ (the boundary of V) then

$$S = \frac{1}{2}\int dV\phi(x)\Box\phi(x).$$

Upon discretization (via a sprinkling) this becomes

$$S_D = \frac{1}{2}\phi_i\Box_{ij}\phi_j$$

where $\Box_{ij}$ is a "local" discrete operator and $\phi_i = \phi(x_i)$ is the value of the field $\phi(x)$ evaluated at the point of the sprinkling which is at x_i.

If $\Box_{ij}$ can be defined in some appropriate way such that Lorentz invariance does not destroy locality, then locality will survive.

Consider the continuum equation

$$\Box_{(x)}G(x,y) = \delta(x,y),$$

which holds for any Green's function of the operator $\Box_{(x)}$. The discretization of this gives

$$\Box_{(x)ij}G_{jk} = \delta_{ik}.$$

This demands that both $\Box_{(x)ij}$ and G_{jk} be non-singular matrices. This work explores the supposition that choosing a non-singular G_{jk} (chosen by discretizing some continuum

Green's function) will give, upon inversion, a matrix G_{ij}^{-1} which will define the discretized operator $\Box_{ij}$.

Before carrying out calculations with sprinklings, the validity of this idea can be tested on regular lattices (which break Lorentz invariance explicitly, but we are here testing for locality only, and so they do provide a valid test of the idea).

As a first example, consider 1-dimensional Minkowski spacetime. The equation

$$\Box_{(x)} G(x,y) = \delta(x-y)$$

becomes

$$-\frac{\partial^2}{\partial t_1^2} G(t_1,t_2) = \delta(t_1 - t_2),$$

which has the solutions

$$G_{ret} = \begin{cases} -(t_1 - t_2), & t_1 > t_2 \\ 0, & t_1 \leq t_2 \end{cases}$$

and

$$G_{adv} = \begin{cases} -(t_2 - t_1), & t_1 < t_2 \\ 0, & t_1 \geq t_2. \end{cases}$$

A non-singular matrix G_{ij} can be formed by discretizing

$$G = \frac{1}{2}(G_{adv} + G_{ret}),$$

yielding

$$G_{ij} = -\frac{1}{2} \mid i - j \mid,$$

which looks like

$$-\frac{1}{2}\begin{pmatrix} 0 & 1 & 2 & \dots & & \dots & N-2 & N-1 \\ 1 & 0 & 1 & \dots & & & \dots & N-2 \\ 2 & 1 & 0 & \dots & & & \dots & N-3 \\ \cdot & & & & & & & \cdot \\ \cdot & & & & & & & \cdot \\ \cdot & & & & & & & \cdot \\ N-3 & \dots & & & \dots & 0 & 1 & 2 \\ N-2 & \dots & & & \dots & 1 & 0 & 1 \\ N-1 & N-2 & \dots & & \dots & 2 & 1 & 0 \end{pmatrix}$$

The inverse G_{ij}^{-1} is:

$$\begin{pmatrix}
\frac{N-2}{N-1} & -1 & 0 & \dots & & & & \dots & 0 & \frac{-1}{N-1} \\
-1 & 2 & -1 & 0 & \dots & & & & \dots & 0 \\
0 & -1 & 2 & -1 & 0 & \dots & & & \dots & 0 \\
\cdot & & & & & & & & & \cdot \\
\cdot & & & & & & & & & \cdot \\
\cdot & & & & & & & & & \cdot \\
0 & \dots & & & \dots & 0 & -1 & 2 & -1 & 0 \\
0 & \dots & & & & \dots & 0 & -1 & 2 & -1 \\
\frac{-1}{N-1} & 0 & \dots & & & & \dots & 0 & -1 & \frac{N-2}{N-1}
\end{pmatrix}$$

The non-local boundary terms $\frac{-1}{N-1}$ in the corners tend to zero as $N \to \infty$, the limit of an infinitely dense lattice.

The remaining (non-boundary) terms are of the form

$$G_{ij}^{-1} = 2\delta_{ij} - \delta_{ij+1} - \delta_{ij-1}.$$

This is certainly a local operator as it has support only on nearest neighbors. It also agrees with what one gets if one discretizes the operator $\frac{-\partial^2}{\partial t^2}$ directly by taking finite differences to represent derivatives:

$$-\frac{\partial^2}{\partial t^2}\phi \to -\frac{\partial}{\partial t}(\phi_{i+1} - \phi_i)$$

$$\to -[(\phi_{i+1} - \phi_i) - (\phi_i - \phi_{i-1})]$$

$$= 2\phi_i - \phi_{i+1} - \phi_{i-1}$$

giving

$$-\frac{\partial^2}{\partial t^2} \to 2\delta_{ij} - \delta_{ij+1} - \delta_{ij-1}.$$

Thus, the idea works well for a regular lattice in 1-dimension when $G = \frac{1}{2}(G_{ret} + G_{adv})$ is used.

Another example is the square lattice (with null edges) in $1+1$ dimensional Minkowski spacetime. In 2-dimensional flat spacetime the equation

$$(-\frac{\partial^2}{\partial t^2} + \frac{\partial^2}{\partial x^2})G(\vec{x}, \vec{y}) = \delta(\vec{x} - \vec{y})$$

has among its solutions

$$G_{ret} = \begin{cases} -\frac{1}{2}, & t_2 > t_1 \text{ and } (\vec{x} - \vec{y})^2 < 0; \\ 0, & \text{otherwise} \end{cases}$$

and

$$G_{adv} = \begin{cases} -\frac{1}{2}, & t_2 < t_1 \text{ and } (\vec{x} - \vec{y})^2 < 0; \\ 0, & \text{otherwise} \end{cases}$$

Once again, a non-singular matrix G_{ij} can be formed by discretizing $G = \frac{1}{2}(G_{adv} + G_{ret})$. The inverse, however, does not obviously appear to be a local operator. The inverse of G_{ij} has been calculated by computer for square lattices as large as 25×25 points. There appears to be no falloff in the matrix entries away from the diagonal, a condition which, if present, would suggest recovery of locality in the dense limit. It is possible that a falloff does occur at a sufficiently slow rate that larger matrices are needed to detect it. Surprisingly, when a discretized Green's function of this form is used for random sprinklings of as few as 200 points, the results accurately reproduce the continuum action for those functions examined, strongly suggesting that the inverse is indeed a local operator. The results of these calculations are presented below.

If one adopts (the more conventional) approach in defining a partial order relation as a reflexive one, i.e. $x \prec x$, then the discretized G'_{ret} will be non-singular. The diagonal will be non-zero, having values $\frac{1}{2}$. [The diagonal value might also be considered a free parameter, as how to define the reflexive condition for partially ordered sets is flexible. In this view, any choice of the diagonal could be considered as the adoption of some "linear combination" of the two definitions of partial orders. Physically, it is a "self-interaction" term which should be negligible compared to the large number of interaction terms present in the matrix $\Box_{ij}$. We examine different values of the diagonal value to test whether the results can be improved for "non-canonical" choice of the parameter.] This non-singular G'_{ret} matrix on the lightcone lattice can be solved explicitly. Its inverse is

$$(G'_{ret})^{-1}_{(ij)(kl)} = 2(\delta_{(ij)(kl)} + \delta_{(i+1j+1)(kl)} - \delta_{(i+1j)(kl)} - \delta_{(ij+1)(kl)}).$$

This is exactly the box operator

$$\sqrt{-g}\ \Box = 2\frac{\partial^2}{\partial u \partial v}, \text{ where } u = x + t \text{ and } v = x - t.$$

discretized by finite differences (u, v are lightcone coordinates).

Thus, these regular lattice calculations suggest that the idea of inverting a discretized Green's function defined on a sprinkling of 1+1 dimensional Minkowski space will give an adequate discretization of the continuum operator $\Box$.

To put this idea to the test, several sprinklings (and so not regular, but random lattices) of an Alexandrov neighborhood in 1+1 Minkowski space were done. The matrices $G_{ret}+G_{adv}$ and G'_{ret} were calculated, inverted, and applied to several different scalar fields. An average over the resulting actions was taken for each scalar field. A match of the average values to the true continuum actions is the criterion under which the conclusion that locality is recoverable in the causal set theory was made. The rest of this section is a step by step discussion of the calculation and the results obtained.

Sprinkling : A Poisson process has the distribution

$$P(N) = \frac{1}{N!}(\frac{N}{V})^N e^{-\frac{N}{V}}$$

for N points to be sprinkled in volume V (at unit density). The first step in the sprinkling was to generate a random N according to this distribution. The parameter V is just the expectation value of N, $V =< N >$, and is a free parameter chosen at the start of the calculation. The random integer N is calculated by elementary Monte Carlo techniques. This part of the computer code is listed in Appendix 1: the routine names are NPTS, PDV, and RAN3 which is taken from the book Numerical Recipes.[5] The seed for RAN3 is calculated in the routine ITIME (also to be found in Appendix 1) based upon the computer's clock.

In 1+1 Minkowski space the measure is uniform (it is $2dudv$). Thus, a sprinkling is easily achieved by generating random numbers x_1, x_2 of uniform distribution on the interval $[0,1] \subset R$ and letting

$$u = 2x_1 - 1,$$

and

$$v = 1 - 2x_2.$$

This yields points sprinkled according to a Poisson process in the region $-1 \leq u \leq +1, -1 \leq v \leq +1$.

N of these points were generated by getting N sets of coordinates (u_i, v_i) in this fashion. The computer code for this is named SPRINKLE, which can be found in Appendix 1. SPRINKLE also calls RAN3, as well as the routine FLD which calculates the field values at each sprinkled point. FLD is described in detail below.

Field Values : As the earlier discussion pointed out, the true action for a scalar field (modulo boundary terms) is

$$S = -\frac{1}{2}\int dV(\vec{\nabla}\,\phi)^2,$$

or equivalently

$$S = \frac{1}{2}\int dV\phi\ \Box\phi.$$

We have heuristic reasons to believe that we can find a discretized $\Box_{ij}$ in causal set terms, possibly modulo boundary terms. Any boundary terms which might get introduced via the discretization are poorly understood at present. For this reason we primarily work with fields which vanish at the boundary. The fields are also chosen to have vanishing gradient at the boundary since the boundary is somewhat "thickened" by the discretization, and since one might not expect boundary contributions to involve higher than first derivatives of ϕ. Some fields that satisfy these criteria are:

$$\phi_1 = [(u^2-1)(v^2-1)]^2,$$

$$\phi_2 = (1+cos\pi u)(1+cos\pi v),$$

$$\phi_3 = (1-u)\phi_1,$$

$$\phi_4 = (1+cos5\pi u)(1+cos5\pi v).$$

Fields ϕ_1 and ϕ_2 are similar in shape, but ϕ_2 has a peak value four times that of ϕ_1 and so it has greater variation over the region. Field ϕ_3 has one peak which is shifted away from the center of the region and so has no symmetry that might give rise to the

correct continuum answer (of zero) for reasons other than locality. Field ϕ_4 is a rapidly varying function which our discretizations should not be able to handle well (not being dense enough) but symmetry might still enforce a "correct" answer in some manner. [It is not clear that an $x \leftrightarrow t$ symmetry can enter so easily into the causal set calculations, though, as causal sets explicitly lack this symmetry. Because of this fact, getting the correct answer of zero (which is a result of the symmetry in the continuum case) could be viewed as a victory in the causal set calculation.]

We also work with linear functions which do not vanish on the boundary. Should any boundary effects be expressed in the inverse Green's function, these fields should detect them. These fields are:

$$\phi_5 = u,$$

$$\phi_6 = v,$$

$$\phi_7 = x \equiv \frac{1}{2}(u+v),$$

$$\phi_8 = t \equiv \frac{1}{2}(u-v).$$

All the above fields have vanishing d'Alembertian action. Fields ϕ_5 and ϕ_6 also have vanishing action for the squared gradient action, but fields ϕ_7 and ϕ_8 do not. In addition, we work with fields that do not have vanishing d'Alembertian action. It is important to use such fields to convince oneself that the method is accurately matching the continuum action, not just always giving a zero result for an entirely different reason. The fields we considered are:

$$\phi_9 = x\phi_1,$$

$$\phi_{10} = t\phi_1,$$

$$\phi_{11} = x\phi_2,$$

$$\phi_{12} = t\phi_2.$$

All of these also have vanishing value and gradient on the boundary. All fields are calculated in the routine FLD which can be found in Appendix 1.

Green's Functions : The starting point for all the Green's functions discussed above is the causal matrix C. Calculating the causal matrix is done in the routine SETC which can be found in Appendix 1. For convenience, we have relabelled the points so that the causal matrix is upper triangular (this is always possible since there are no closed timelike curves). This is accomplished in the routine RELAB which can be found in Appendix 1.

The now upper triangular C is easily seen to be 2 G_{ret}. If we add it to its transpose we have

$$2(G_{ret} + G_{adv}) = C + C^T.$$

$C+C^T$ is generated with routine GSYM which can be found in Appendix 1.

Adding the unit matrix δ to C gives $2G'_{ret}$:

$$2G'_{ret} = \delta + C.$$

The diagonal parameter in G'_{ret} is studied by using $2G'_{ret}(\lambda) = \lambda\delta + C$. This is done in the routine GRET which can be found in Appendix 1.

G'_{ret} is automatically nonsingular. This is not true for $G_{ret}+G_{adv}$, however. It frequently happens that two or more rows will be identical in $G_{ret}+G_{adv}$. This occurs as a result of the corresponding points being sprinkled closely together, but spacelike separated, as the points p_1 and p_2 in Figure 5 below.

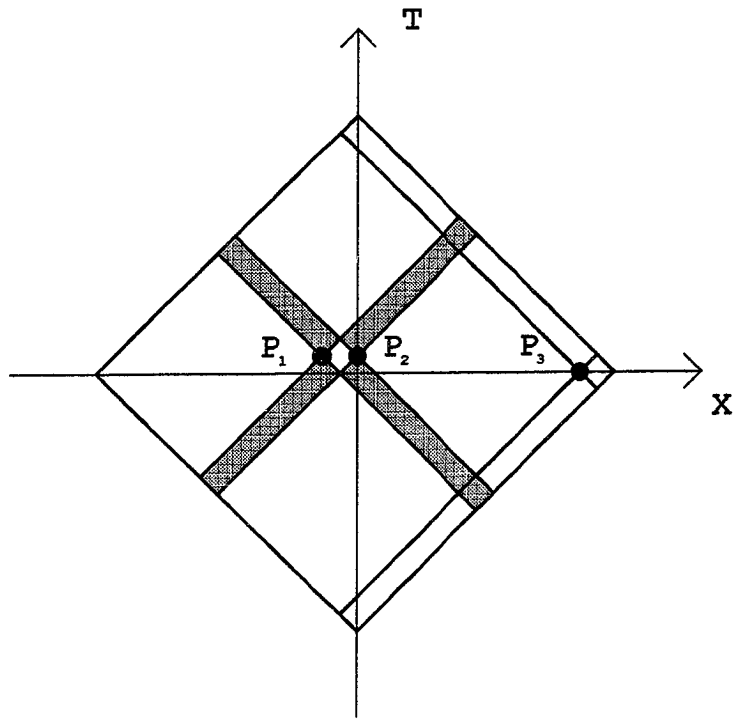

Figure 5

The volume of the shaded region is smaller than one and so no points are likely to have been sprinkled there. Then

$$J^+(p_1) = J^+(p_2),$$

and

$$J^-(p_1) = J^-(p_2).$$

In causal set terms we can view such pairs as being indistinguishable, or "non-Hegelian" to use a term coined by Sorkin. Such a pair is said to comprise a "clique" in the theory of partially ordered sets.

Another (much rarer) source of singularity is when a point (indicated by p_3 in the above figure) is so close to the x-axis corners of the region that the intersection of its lightcone with the region has volume smaller than one. Then

$$J^+(p_3) = \emptyset,$$

and

$$J^-(p_3) = \emptyset.$$

This gives rise to a row of zeros in $G_{ret}+G_{adv}$. Such points might be more properly thought of as belonging on the other side of ∂V, having been located here within V due to the thickening of the boundary caused by the discretization.

The first source of singularity is dealt with by merging the two points into one, located at one of the two original points (i.e. one is deleted; the one which gets deleted is an artifact of the order in which they were originally sprinkled). This is done in the routine DEL2 which can be found in Appendix 1. The second source of singularity is dealt with by pushing the point over the boundary where it "more properly belongs" (i.e. it is deleted). This is done in the routine DISCARD1, which can be found in Appendix 1.

Inversion : Once the Green's functions have been computed, they are inverted by the routine INV, which bridges between calling program POBOX and the routines LUDCMP and LUBKSB, which are taken from the book Numerical Recipes.[5] These can be found in Appendix 1. As these routines do not catch "nearly singular" (or even some exactly singular!) matrices very well, the inverse is multiplied by the original matrix and the result is checked for its nearness to the unit matrix. Those matrices which do not invert well are discarded.

Calculation of the Action : The inverse matrix is the candidate $\Box_{ij}$. This is applied to each of the fields (described above) in turn. The calculation is performed in the routine ACT, which can be found in Appendix 1.

Gathering Statistics : The routine MAIN (found in Appendix 1) calls the routine POBOX which in turn calls the routines described above. MAIN collects the results of all the runs and calculates running averages, standard deviations, and standard errors.

Results : The results from calculations on a VAX computer are presented in Tables 1 and 2. Table 1 presents results for the symmetrized Green's function. ϕ_3 was calculated by averaging 298 runs. The other results are for averaging a different set of 235 runs. All results are for sprinklings with $< N >= 200$. The first column of Table 1 lists the field, the second column lists the continuum value of $\int dV \phi \Box \phi$, with the value of $-\int dV (\nabla\phi)^2$ given in parenthesis for ϕ_7 and ϕ_8. The third column gives the calculated average and

standard error of $\phi_i G_{ij}^{-1} \phi_j$.

Field	$\int dV \phi \Box \phi$	$\phi_i G_{ij}^{-1} \phi_j$
$\phi_1 = [(u^2-1)(v^2-1)]^2$	0	$.04 \pm .10$
$\phi_2 = (1+\cos \pi u)(1+\cos \pi v)$	0	$.4 \pm .52$
$\phi_3 = (1-u)\phi_1$	0	$.003 \pm .081$
$\phi_4 = (1+\cos 5\pi u)(1+\cos 5\pi v)$	0	8 ± 52
$\phi_5 = u$	0	1.7 ± 2.0
$\phi_6 = v$	0	1.4 ± 3.5
$\phi_7 = x$	0 (−2)	-3.0 ± 1.8
$\phi_8 = t$	0 (+2)	4.5 ± 1.2
$\phi_9 = x\phi_1$	.1651	$.161 \pm .028$
$\phi_{10} = t\phi_1$	-.1651	$-.173 \pm .019$
$\phi_{11} = x\phi_2$	2.25	$2.13 \pm .36$
$\phi_{12} = t\phi_2$	-2.25	$-2.23 \pm .26$

Table 1. Symmetric Green's Function

The calculated values agree quite well with the continuum values. The agreement is to within a single standard error for all fields except ϕ_7 and ϕ_8 (ϕ_7 agrees to two standard errors but ϕ_8 does not agree). However, these agree much better with the action $-\frac{1}{2}\int dV(\vec{\nabla}\,\phi)^2$ (which has continumm value of -2 and $+2$ respectively). Since boundary terms are suspected to be present in ϕ_7 and ϕ_8, we did not expect agreement here. Also, we did not expect agreement for ϕ_4 as the function has rapid variation. This seems to be reflected in the comparatively large standard error. Those fields which have non-zero continuum action have calculated actions that agree with the continuum actions, but do not agree with zero.

Table 2 presents results for the retarded Green's function. $< N >= 200$ for all the data. The three final columns list the results for various values of the diagonal parameter λ. The value $\lambda = 1$ is the "canonical" choice, and the results are for 300 runs. The results

for $\lambda = \frac{3}{4}$ are for 262 runs. The results for $\lambda = \frac{1}{2}$ are for about 240 runs, the actual number of runs varying slightly with each field because some grossly large data were thrown out (meaning that the program initially had a large cutoff value of 1000 for the action of each field, presupposing that any actions calculated to be higher than this were due to nearly singular matrices, which are not inverted accurately, and so the data was thrown out as being innacurate on these grounds).

Many of the results for the retarded Green's function are poor, and none are as good as for those of the symmetric Green's function. The results may be poor since effectively only half the interactions as those of the symmetric Green's function are included. Small numbers of runs for larger numbers of points have been performed. The results do improve as $< N >$ increases, however the standard deviations seem to grow with $< N >$.

Field	$\int dV \phi \Box \phi$	$\phi_i G'^{-1}_{ij} \phi_j$		
		$\lambda = 1$	$\lambda = \frac{3}{4}$	$\lambda = \frac{1}{2}$
$\phi_1 = [(u^2-1)(v^2-1)]^2$	0	$-.073 \pm .025$	$-.15 \pm .20$	-4 ± 15
$\phi_2 = (1+\cos\pi u)(1+\cos\pi v)$	0	$-1.45 \pm .30$	-0.7 ± 2.1	-40 ± 39
$\phi_3 = (1-u)\phi_1$	0	$-.099 \pm .026$	$-.28 \pm .20$	-28 ± 14
$\phi_4 = (1+\cos 5\pi u)(1+\cos 5\pi v)$	0	39 ± 25		
$\phi_5 = u$	0	-5.0 ± 9.2	35 ± 42	
$\phi_6 = v$	0	-6.5 ± 8.9	$15 \pm 40.$	
$\phi_7 = x$	0(−2)	$1.03 \pm .24$	-5.3 ± 3.4	
$\phi_8 = t$	0(+2)	-6.7 ± 8.7	-13 ± 42	
$\phi_9 = x\phi_1$	.1651	$.0634 \pm .0013$	$.0636 \pm .0079$	$1.03 \pm .68$
$\phi_{10} = t\phi_1$	−.1651	$-.1108 \pm .0062$	$-.006 \pm .067$	8 ± 11
$\phi_{11} = x\phi_2$	2.25	$.866 \pm .016$	$.907 \pm .083$	8.9 ± 5.7
$\phi_{12} = t\phi_2$	−2.25	$-1.476 \pm .064$	$-.70 \pm .57$	19 ± 27

Table 2. Retarded Green's Function

Since the results for the symmetrized Green's function reproduce the continuum val-

ues well, it can be concluded that locality is recoverable.

Finally, some calculations using the symmetric Green's function were done to test the variation of the standard deviation (σ) with $< N >$ for some of the fields. In these calculations, the number of points was actually fixed (as opposed to varying randomly about $< N >$). The values of N were 16, 32, 64, 128, 256, 512, 700, and 1024. For each N, the standard deviation was calculated from 42 trials. The results are given in Table 3. The first column lists the field. The second column lists the correlation coefficient between $\log N$ and $\log \sigma$. The third column lists the slope α, where $\sigma \sim N^{-\alpha}$. In all cases the standard deviation decreases faster than N^{-1}.

Field	correlation coefficient	α
$\phi_2 = (1 + \cos \pi u)(1 + \cos \pi v)$	.78	1.8
$2\phi_{11} = 2x\phi_2$	.995	1.6
$2\phi_{12} = 2t\phi_2$	.995	1.65

Table 3.

Part IV

Enumerating Causal Sets

An interesting problem is that of enumerating the partially ordered sets definable on N elements (D_N). The answer is known for N≤9. For N = 10, there is a dispute.[6,7] A value for $N = 11$[7] is given, but it is not guaranteed to be correct until the dispute over $N = 10$ is resolved.

Culberson and Rawlins[7] have empirically found some tantalizing patterns in the values n(R,N), the number of posets on N elements that contain R relations. They conjecture that these patterns, once understood, may lead to a solution of the general problem of enumeration.

The causal set program is interested in *connected* posets. Appendix 2 lists a computer program that calculates C_N, the number of connected posets definable on N elements. The program also calculates n_C(R,N), the number of connected posets of N elements that contain R relations, and m_C(L,N), the number of connected posets of N elements that contain L links. The algorithm used will be described in detail at the end of this section.

Given the set of values $\{D_N\}$ one can deduce the set of values $\{C_N\}$ and vice versa. D_N is a function of $\{ C_M \mid M \leq N \}$. The function is straightforward to calculate. To do so, begin by listing all partitions of N. This gives the set

$$\{\{I_j^{p_j} \mid I_m \neq I_n \quad \text{if} \quad m \neq n, \quad \text{and} \quad \sum_j p_j I_j = N\}\},$$

where the notation I^p means p copies of the integer I.
(For example, N=4 has the partitions $\{\{4\},\{3,1\},\{2^2\},\{2,1^2\},\{1^4\}\}$.)

Then

$$D_N = \sum_{partitions} \prod_j \binom{C_{I_j} + p_j - 1}{p_j},$$

where $\binom{M}{N}$ is a binomial coefficient. For example

$$D_4 = \binom{C_4}{1} + \binom{C_3}{1}\binom{C_1}{1} + \binom{C_2+1}{2} + \binom{C_2}{1}\binom{C_1+1}{2} + \binom{C_1+3}{4}.$$

Given the set {n(R,N)} one can likewise deduce the set { n_C(R,N)} and vice versa. The calculation is similar to that above with the added complication that within each partition of N one must sum over the partitions of R.

For N$\leq$ 7 the values calculated by the program agree with the accepted results. The values m_C(L,N) are listed in Table 3. The values n_C(R,N) are listed in Table 4. From this data can be deduced the values of n(R,N) which are listed in Table 5.

The patterns in n(R,N) are polynomials, which is what one expects if they are determined from simple combinatoric functions. It is possible that the values n_C(R,N), or even m_C(L,N) exhibit patterns which are simpler than those of n(R,N). Further investigation may reveal the source of these patterns (assuming they exist) and thereby solve the problem of enumeration. However, the work of Dhar[8] might indicate that there is an infinite heirarchy of such patterns and so the solution may be much more elusive.

As there is some dispute over the value of D_{10}, the program may also give an independent check upon this value. However, the VAX does not have the necessary speed to go beyond N= 7. Work is proceeding to adapt the code to faster computers and to speed up the algorithm.

What follows is a detailed description of the algorithm used to enumerate causal sets in the program GENPOS, which is listed in Appendix 2. The program works with order preserving labellings. An order preserving labelling of a causal set C is an assignment of integers to the points of the causal set so that if two points $p, q \in$ C are such that $p \prec q$, then the integer assigned to p is smaller than that assigned to q. With such a labelling, both the link and causal matrices representing C are upper triangular. For a causal set consisting of N elements, there are $\frac{N(N-1)}{2}$ potential links that might exist between the N elements. For a causal set to be connected, there must be at least $N-1$ links. There can be at most $(\frac{N}{2})^2$, if N is even, or $(\frac{N-1}{2})(\frac{N+1}{2})$, if N is odd, links in a causal set of N elements. Within these limits the algorithm forms candidate link matrices and tests them for validity and uniqueness.

The first step is to place a given number of links into an upper triangular matrix to form a candidate link matrix. The variable *nlinks* is the number of links. Using the routines NEXTLINK, ISUM, and L, a candidate link matrix consisting of *nlinks* links is obtained. NEXTLINK ensures that all combinations of *nlinks* links are tried. With the candidate link matrix (called *lorg*), a causal matrix (called *causal*) is calculated by transitive closure. From *causal* a true link matrix (called *lmat*) is calculated. If the candidate does not match the true matrix it is discarded and the next candidate is generated.

When a good link matrix is found, it is first checked to insure that the represented causal set is connected. If disconnected, the matrix is discarded and a new candidate is generated. If the causal set is connected, a matrix of data (called *nlf*) is formed based upon the number of points linked to and causally related to each point of the causal set. From this data another matrix (called *ninv*) is formed which is invariant under relabellings of the causal set.

The matrix of invariants is used to compare with previously saved causal sets which are stored in files. Two of the invariants are used to direct the saved sets into different files to speed the comparison process. The two invariants used are the number of minimal and maximal elements of the causal set. If no matching invariant matrix is found, the current causal set is added to the end of the appropriate file. When a matching invariant matrix is found, the current causal set must be checked for a relabelling which makes it match the saved causal set. An initial relabelling is done which matches the matrix *nlf* to the corresponding matrix of the saved set. If the link matrices match the causal set is discarded. If not, the matrix *nlf* is used to reduce the permutation group S_N (corresponding to all relabellings) to a subgroup which is then gone through using the routine NEXTPERM until a match is found. A match results in the discarding of the causal set. If no match is found with any of the saved causal sets in the appropriate file, the causal set is added to the end of the file.

After all possible matrices of the given number of links is tested, all accumulated causal sets are erased and the process begins again with causal sets consisting of one link more than those of the previous loop. As each new causal set is found, the number of relations is counted and n_C(R,N) is updated, as well as m_C(L,N) and C_N. At the end of

the program these results are placed into a file called *genpos.dat*. To start the program for any N requires the parameter *npts* in the first line of the program to be set to the value N.

	N=1	2	3	4	5	6	7
L=0	1						
1		1					
2			3				
3				8			
4				2	27		
5					12	91	
6					5	87	350
7						45	532
8						12	475
9						3	201
10							71
11							14
12							7

$\mathbf{m_C(L,N)}$

Table 3. Number of connected posets on N elements with L links.

	N=1	2	3	4	5	6	7
R=0	1						
1		1					
2			2				
3			1	3			
4				3	6		
5				3	7	10	
6				1	11	20	22
7					9	33	50
8					6	40	107
9					4	40	158
10					1	34	204
11						28	220
12						17	222
13						10	194
14						5	163
15						1	122
16							83
17							54
18							29
19							15
20							6
21							1
C_N	1	1	3	10	44	238	1650

$n_C(R,N)$

Table 4. Number of connected posets on N elements with R relations.

	N=1	2	3	4	5	6	7
R=0	1	1	1	1	1	1	1
1		1	1	1	1	1	1
2			2	3	3	3	3
3			1	4	6	7	7
4				3	10	16	18
5				3	10	25	38
6				1	12	36	74
7					9	43	113
8					6	46	167
9					4	44	209
10					1	35	243
11						28	249
12						17	239
13						10	204
14						6	168
15						1	123
16							83
17							54
18							29
19							15
20							6
21							1
D_N	1	2	5	16	63	318	2045

n(R,N)

Table 5. Number of posets on N elements with R relations.

Part V

Open Questions and Speculations

Quantization of the causal set theory will require the specification of an action principle and a method of coarse graining which satisfy the requirement that when a sum over all causal sets on N elements is performed (with appropriate boundary conditions), the dominant contribution must come from those causal sets which, upon coarse graining, agree with the continuum geometry of classical gravity. [The geometry of classical gravity is ordinarily assumed to be that of a manifold. This assumption can be expected to break down at singularities. In addition, a space that is a cobordism mediating between different spatial geometries (as in pair creation of topological geons) might arise in the sum with some additional phase arising from the degeneracy of the metric at the points of topology change.]

Carrying out this program will involve several details which have not to date been worked out. This section is devoted to a discussion of some of these details, particularly the form of the action, techniques for carrying out the sum over causal sets, and the inclusion of quantum fields. Also, there is a brief discussion of why topological spin-statistics theorems might be extended to the discrete realm of causal sets.

Action Principle : The primary issue is of course the form of the action principle. It should consist of an "action functional" defined upon causal sets such that a course graining of those sets which dominate the sum over causal sets will be faithfully embeddable into manifolds. Quantum amplitudes will be gotten by exponentiating this action. Thus, given a causal set C, an action S[C] is defined, giving an amplitude

$$A[C] = e^{iS[C]}.$$

The exact rules for forming probabilities are not yet known for quantum gravity in general, and causal sets in particular. Formally, one can define a "quantum expectation value" for any property by a sum over causal sets weighted by the amplitude. For example, if one desires the expectation value of the causal dimension, $\dim_C$, of a causal set C (the causal dimension of C is calculated in a manner so that if C is faithfully embeddable into a manifold M, then $\dim_C$=dim M; c.f. ref. 3), it is given by

$$< \dim_C > = \frac{\sum_C \dim_C(C) e^{iS[C]}}{\sum_C e^{iS[C]}}.$$

[Notice that there is no measure term. In the usual sum over histories quantization of a classical theory, the action S is the classical action and the measure is present to make the expectation values come out correctly. One might just as well define a *quantum* action by

$$S_Q[\phi] = S_{Cl}[\phi] - i \ln \mu[\phi],$$

and then the sum over histories equation for expectation values will not have a measure term. The weights will be exponentiated *quantum actions*, which are the classical actions with a quantum correction. In this view, the measure just encodes quantum corrections to the classical action. Since there is no classical action to be corrected in the causal set theory, it will not be necessary to separate off part of S and call it a measure.]

Many forms for the action S[C] have been proposed. Little work has been done to explore the viability of any of these actions. To give a flavor of the ideas, three proposals will be discussed.

One proposal, due to Geroch, is that S[C]=2πZ[C], where Z[C] is a functional on causal sets C such that it is integer valued when C is faithfully embeddable into a manifold and not integer valued when C is not faithfully embeddable. Then

$$\sum_C e^{iS[C]}$$

will have constructively adding contributions for faithfully embeddable causal sets. If the spectrum of Z[C] is such that there exists C^* satisfying

$$Z[C^*] = Z[C] \pm \frac{1}{2},$$

for every non embeddable C, then non embeddable causal sets will cancel.

A second proposal, due to Sorkin, is to define S[C] to be the logarithm of the number of trees spanning C. This is calculable using the matrix tree theorem. The calculation is as follows. Given L, the link matrix of a causal set, one first defines the diagonal matrix D by

$$D_{ij} = \delta_{ij} \sum_k L_{kj}.$$

Then one defines the matrix M by

$$M \equiv D - L.$$

Finally, $M^{(0)}$ is defined as a matrix derived from M by deleting a column consisting of all zeros and its corresponding row. Then the determinant of $M^{(0)}$ is the number of trees rooted at the point corresponding to the deleted row and column. This definition excludes any spanning subgraph that doesn't literally look like a tree (with no branches dipping downward). The determinant is easily calculated in this case as the matrix $M^{(0)}$ can be made upper triangular (with diagonal entries only being permuted) and so the result is the product of the diagonal entries (excluding the zero to be deleted). [Thus, the determinant of D (disregarding the zero to be deleted) also gives the correct number of spanning trees.] By the construction of the matrices D and $M^{(0)}$ it is easily seen that the number of spanning trees is equal to the product of the number of downward links from each point above the root point.

This idea derives from the intuitive picture of continuum expanding geometries (e.g. the FRW model) as spacetime consisting of a partial history "growing like a tree" (i.e. the new portion of spacetime arises locally).

A third proposal, due to Meyer, is to define S[C] to be some multiple of the Euler number of C. This has the advantages that it is plausibly related to the Euler number of a manifold into which C can be embedded when such an embedding is possible (and therefore should constructively interfere), and that it is very easily calculated. Given any causal set C, one can associate to it a simplicial complex $\Delta(C)$ by mapping elements and relations of C to vertices and edges in $\Delta(C)$. The euler number of C is defined to be the

euler number of $\Delta(C)$:

$$\chi(\Delta(C)) = \sum_{k=0}^{\infty}(-1)^k c_k,$$

where c_k is the number of $k+1$ element chains of C. A more efficient calculation is possible by calculating the Möbius function of $\hat{C}$, which is the causal set defined by attaching minimal and maximal elements ($\hat{0}$ and $\hat{1}$) to C. The Möbius function is

$$\mu_{\hat{C}}(\hat{0},\hat{1}) = \sum_{k=0}^{\infty}(-1)^k \hat{c}_k,$$

where $\hat{c}_k$ is the number of $k+1$ chains from $\hat{0}$ to $\hat{1}$ in $\hat{C}$. The Möbius function satisfies the recursion relation

$$\mu(x,y) = -\sum_{x \preceq z \prec y} \mu(x,z),$$

when $x \prec y$ in a causal set. This recursion relation allows an efficient calculation of μ_C as one works upward along links of $\hat{C}$ and accumulates a sum over preceding points, in the end the sum will be the Möbius function. Finally, the identity

$$\chi(\Delta(C)) = 1 + \mu_{\hat{C}}(\hat{0},\hat{1})$$

yeilds the euler number of C.

All these actions take their values in a discrete space. This prevents the usual interpretation of classical theories arising as the stationary phase approximation to the quantum theory. This view could be retained, however, if S[C] admits a supremum at a collection { C_{Cl}} of C's and varies rapidly amongst those C's which are not in the collection { C_{Cl}}. Then it is required that { C_{Cl}} primarily consists of causal sets which are faithfully embeddable into manifolds.

An interesting variation of the approach of defining an action for each causal set is to work with the partial order of partial orders. The elements of this partial order are all possible partial orders. The relation can be defined as $C_1 \prec C_2$ if C_1 is a subposet of C_2. If a path is defined as an unbroken chain of links, then one can define a transition amplitude between two causal sets as a sum over paths of a function of each path, the function taking as argument a product of elements of a group attached to each link of the path.

For example, if the group is the set of nonzero real numbers with ordinary multiplication as product and the identity element (i.e. the number 1) attached to each link, then the transition amplitude between two causal sets is the number of paths connecting them in the partial order of partial orders. In this approach the transition amplitudes automatically satisfy the important condition

$$A(C_1 \to C_2) = \sum_{C_3} A(C_1 \to C_3) \times A(C_3 \to C_2),$$

where $C_1 \prec C_3 \prec C_2$ and all C_3's which are summed over must be elements of a complete anti-chain (i.e. no C_3 is related to any other C_3 and there is no other causal set which can be added and still have an anti-chain).

Calculating : As mentioned, no calculations have been done to test the viability of any of the actions that have been proposed. There are technical difficulties which need to be addressed before such calculations can be performed with any confidence.

One difficulty is in carrying out the sum over all causal sets. The number of causal sets definable on N elements grows very rapidly with N. Asymptotically the number is :

$$C \times 2^{\frac{N^2}{4}+\frac{3}{2}N} e^N N^{-(N+1)}, \quad \text{where} \quad C \approx .8.$$

There is no exact expression for finite N. Further discussion of this problem is given in Part IV.

Since the number of causal sets is large, calculations cannot cover the full sum (although the so-called "generic" causal sets are so simple in structure that *thier* contributions to the sum might be exactly calculable in some cases). Some technique must be used to efficiently approximate the sum with a sum over a much smaller set of causal sets. Many techniques exist to find the maximal action and sum over the neighborhood of those points which maximize the action. An example is the technique of simulated annealing. These techniques may require some type of analytical continuation to real amplitudes, however.

It is important in such schemes that all points are equally likely to be reached in a random walk when thier actions agree. It is also important that all moves which are used to wander through the space of causal sets must be reversible, so that the algorithm cannot get trapped at a particular set. It is possible to define such moves.

A causal set is represented by its link matrix L (which has $L_{ij} = 1$ if there is a link connecting points labelled by numbers i and j and zero otherwise), and its causal matrix C. One move is the addition of a link ℓ_{ij}. This can only be done if $C_{ij} = 0$, and $[(\delta+C^T)L(\delta+C^T)]_{ij} = 0$. Then the new C is the transitive closure of the new L. The reverse of this move is the deletion of an existing link from L, the new causal matrix again being gotten by transitive closure of the new L. It must be insured that the deletion does not cause the set to become disconnected. A causal set of N elements is connected if $(\delta + C + C^T)^{N-1}$ has no zero entries. These moves are reversible. However, there are causal sets which cannot be reached by these moves (e.g. the completely ordered set which can have no link added or removed because of the conditions of these rules). It is easy to get around this problem by including disconnected causal sets in the calculation, giving them zero weight (as most causal sets are connected, this will not harm the efficiency of the calculation). Then all causal sets will be reachable by the reversible steps.

There remains the problem that different matrices which represent the same causal set must be recognized. Given complete freedom of relabelling, the labellings are in 1–1 correspondence with elements of the permutation group, S_N, for a causal set consisting of N elements. Not all relabellings give rise to distinct matrices, as the automorphism group of a causal set, $\mathcal{A}_C$, will leave unchanged any matrix representing that set. The matrices that represent C are in 1–1 correspondence with elements of the set $S_N/\mathcal{A}_C$ (which unfortunately is not a group as $\mathcal{A}_C$ is not in general a normal subgroup of S_N). One can handle the different matrices representing C by weighting each by the number of these matrices (which is order$[S_N/\mathcal{A}_C]$) and performing a sum over the space of causal matrices, $\mathcal{M}_C$; i.e. the sum over causal sets is replaced by a sum over causal matrices:

$$\sum_C e^{iS[C]} \to \sum_{\mathcal{M}_C} \frac{1}{r_C} e^{iS[C]}, \quad \text{where} \quad r_C = \text{order}[S_N/\mathcal{A}_C].$$

This method may be inefficient, as the number of matrices representing any given causal set will be very large. One can dramatically improve the efficiency by restricting to, say, upper triangular matrices (any causal set will have at least one such representation). Now the relabellings do not form a group, but this probably presents no problem. Further

improvements in efficiency might be possible. This question has not yet been looked into. It is clearly important to the issue of carrying out the sum over causal sets in practice.

Field Theories : It has been suggested[9] that coarse graining provides a natural mechanism via which fields will emerge from the causal set theory. Consideration of this idea suggests that the usual gauge field theories of the Standard Model might arise in this fashion.

A gauge field theory begins with an action that is invariant under the action of a group G. This symmetery is broadened to that of a gauge symmetry by demanding that the action be invariant under independent actions of the group G at each spacetime point. Under the action of this enlarged group, the usual kinetic term of the action will acquire extra terms involving a new field which takes its value in the Lie algebra g of G at each spacetime point. These terms are cancelled by the introduction of a connection A of a principle fiber bundle (M, G) of the group G over the manifold M. The connection is defined by giving its value $A(x)$ at each point of the manifold $x \in$M. $A(x)$ is called the gauge field associated with the gauge symmetry. (To complete the now gauge invariant action, a kinetic term for the gauge field is added; the square of the curvature associated with the connection A.)

The dynamics of the gauge field, then, relies upon the algebra g. This algebra is the same for any group G/D, where D is a discrete normal subgroup of G. For example, $G_1 = SU(2)$ and $G_2 = SO(3) = SU(2)/Z_2$ will have the same associated $su(2)$ gauge field.

Consider a coarse grained causal set C. All causal sets, which when coarse grained give rise to C, will contribute to the sum over histories contribution of C. In particular, when C is faithfully embeddable into a manifold M, there are a large number of causal sets related to C which will contribute to the amplitude for M. Among such contributing causal sets will be those consisting of replacing some links in C by some binomial partial order. (Figure 2 on page 5 is an example of replacing each link of a causal set embeddable into 1-dim Minkowski space with B_2.) As discussed in Part II, the binomial poset B_N is based upon the $N-1$ simplex. As the starting causal set C embeds into a manifold, the replacement of a link in C (for this discussion the link is assumed to be embedded into a small region of the manifold as defined by the manifold's curvature) by the subposet B_N

can be viewed as the attatchment of an $N-2$ sphere to the manifold with the $N-1$ simplex embedded symmetrically in this sphere. [The $N-2$ sphere, and not the discrete structure B_N directly, is attached to the manifold so that the classical manifold picture is preserved. The resulting manifold is locally similar to a Kaluza-Klein type space.] The $N-1$ simplex has the discrete symmetry group A_N, the alternating group of order N. In this way the group $SO(N-1)/A_N$ might get attatched to each point of the manifold M. The associated Lie algebra connection is then $so(N-1)$-valued. In particular, $so(3) = su(2)$ and $so(2) = u(1)$.

In this fashion the familiar gauge fields may be recoverable as a basic feature of coarse graining. There are, of course, many other causal sets which give C upon coarse graining. It is possible that other fields may arise from these. It may be possible that these causal sets do not contribute to the sum over histories. For instance, consider a causal set C_1 which has a subposet C_s which will be replaced by a link upon coarse graining, yielding the set C. Consider another causal set C_2 which has a subposet TC_s, the time reverse of C_s, which also will be replaced by a link upon coarse graining, yielding the same set C. An action having the property

$$S(C_2) = S(C_1) \pm \pi$$

for such related causal sets C_1 and C_2, has the feature that only time-symmetric subposets will contribute to the sum over histories. Similar conditions may further limit the number of contributing causal sets. The point is that highly symmetric subposets will survive such "paring conditions". Such considerations might even prove helpful in defining a suitable action principle, since the known gauge fields should result from the causal set theory.

Spin-Statistics Theorems : Spin-Statistics theorems play an important role in current physical theories. Theorems have been proven using topological notions which rely upon the continuous nature of configuration spaces.[10-14] Field theory proofs rely upon the algebra of field operators; presumably this algebra embodies the topological properties which are the true source of the association between spin and statistics.

The theory of causal sets, being discrete, cannot incorporate the topological proofs, except in a continuum approximation. Perhaps there is some more fundamental geometric property which is responsible for the association that both discrete and continuum sets

share.

Statistics involves the behavior of a system under exchange of two identical elements of that system. For greater than 2+1 dimensions, if exchange is denoted by E, then $E^2=1$, the identity transformation on the system. In classical mechanics this is all that may be said. In quantum mechanics the indistinguishability of the objects leads to the further relation $E=\pm 1$. Thus, statistics has two algebras: 1 for bosons and Z_2 for fermions.

Spin involves the behavior of a system under a 2π rotation of an element of that system. Classically $R(2\pi) = 1$, but quantum theory allows the possibility of $R(2\pi) = \pm 1$, $+1$ being for tensorial objects and -1 for spinorial objects. The key question is what geometric feature allows the spinorial possibility.

The proof that $R(4\pi) = 1$ is a homotopy between a 4π rotation and no rotation at all. Dirac's scissors[15] is a familiar demonstration of this homotopy. Another demonstration is that given in Misner,Thorne,and Wheeler.[16] It involves the rotation of an object which is tied to a supporting structure which is *not* rotated. The 4π rotation can be continuously deformed to no rotation without further rotation of the object. (This relative rotation also plays a key role in the discussion of spin properties of topological geons.[17]) Thus $R^2(2\pi) = 1$ is guaranteed. [There is no homotopy between the 2π rotation and the identity.]

If in this example the ties to the supporting structure are cut, a 2π rotation followed by reattachment will reproduce the same configuration. This suggests the proposition that it is the connectivity of the relevant structures that is important to understanding spin. Furthermore, a discussion of which manifolds support spinor structures emphasizes the global connective structure of the manifold.[18] Continuous topology is in fact precisely the structure which encodes such connectivity information. Thus, it seems probable that this is the feature which is key to the spin-statistics correlation.

The causal set theory does have connectivity information. At first glance this may seem to be untrue, as the relevant connective structure is that of spacelike hypersurfaces of a manifold. The analogous structure in the causal set theory is that of an anti-chain, a set of points with no two points related. Such a collection superficially looks like a "dust", a collection of points with no structure. However, the points still have all their relations, which define a structure. In particular, the links to these points can play the role of the

connecting ties that are required to demonstrate $R(4\pi) = 1$. It might be objected that links are not spacelike ties. This is not a real objection, as all that is important in the homotopy is that there is a rotated region tied to an unrotated region.

[This discussion is only meant to be suggestive. How to rigorously exploit the connective structure of causal sets to give a spin-statistics theorem when homotopies are undefinable in the traditional sense is not immediately apparent.]

The association between spin and statistics depends upon two logically distinct features: (i) the mysterious fact (first noted by Feynman[19]) that antiparticles can be thought of as particles travelling backward in time, and (ii) the phenomena of creation and annihilation of particle-antiparticle pairs. It is one of the ingenious features of Feynman diagrams that they combine these features to appear naturally as two aspects of the same phenomenon. Together, these two features yield a spacetime homotopy between 2π rotation of a single particle and exchange of two identical particles.

An interesting question is if the continuum proofs can be extended to causal sets.

REFERENCES

1) R.P. Stanley [1986],*Enumerative Combinatorics:Volume I*, (Monterey:Wadsworth & Brooks/Cole)chapter 3.

2) P.C. Fishburn [1985],*Interval Orders and Interval Graphs: A study of partially ordered sets*,(New York:Wiley).

3) D.A. Meyer [1988],*The Dimension of Causal Sets*, Ph.D. Thesis, Massachusetts Institute of Technology.

4) L. Bombelli, J. Lee, D. Meyer and R.D. Sorkin [1987],"Space-time as a causal sets", Phys. Rev. Lett. 59, 521–524.

5) W.H. Press, B.P.Flannery, S.A. Teukolsky and W.T. Vetterling [1986], *Numerical Recipes: The art of scientific computing*, (Cambridge:Cambridge University Press)chapter 2 for inversion, chapter 7 for random number generator.

6) R.H. Möhring [1984], "Algorithmic aspects of comparability graph and interval graphs", in Graphs and Order; *The Role of Graphs in the Theory of Ordered Sets and Applications (ed. I. Rival)*, Volume 147, pages 41–102. NATO Adv. Study Inst. Ser. C: Math. Phys. Sci., Vol. 147.

7) J.C. Culberson and G.J.E. Rawlins [1991], "New results from an algorithm for counting posets", Order 7, 361–374.

8) D. Dhar [1978], "Entropy and phase transitions in partially ordered sets", J. Math. Phys. 19, 1711–1713.

9) L. Bombelli [1987],*Space-time as a Causal Set*, Ph.D. Thesis, Syracuse University.

10) D. Finkelstein and J. Rubinstein [1968], "Connection between spin, statistics, and kinks", J. Math. Phys. 7, 1762.

11) R.D. Sorkin [1988], "A general relation between kink-exchange and kink-rotation", Commun. Math. Phys. 115, 421–434.

12) R.D. Tscheuschner [1989], "Towards a topological spin-statistics theorem in quantum field theory", Int. J. Th. Phys. 28, 1269 [Erratum in Int. J. Th. Phys. 29(1990) 1437];J. Math. Phys. 32(1991)749.

13) A.P. Balachandran, A. Daughton, Z.-C. Gu, G. Marmo, R.D. Sorkin, and A.M. Srivastava [1990], "A topological spin-statistics theorem or a use of the antiparticle", Mod. Phys. Lett. A, 5,1575; Physica Scripta *T36*(1991)253.

14) A.P. Balachandran, A. Daughton, Z.-C. Gu, G. Marmo, R.D. Sorkin, and A.M. Srivastava [1992], "Spin-statistics theorems without relativity or field theory", Syracuse University preprint SU-4228-433.

15) R. Penrose and W. Rindler [1984], *Spinors and Space-Time: Volume I*, (Cambridge University Press:Cambridge)chapter 1.

16) C.W. Misner, K.S. Thorne and J.A. Wheeler [1973],*Gravitation*,(San Francisco:W.H. Freeman).

17) R.D. Sorkin [1986], "Introduction to topological geons", in *Topological Properties and Global Structure of Space-time: Proceedings of the International School of Cosmology and Gravitation, Erice*,1985,eds. P.G. Bergmann and V. de Sabbata, (New York:Plenum) 249–269.

18) R.M. Wald [1984],*General Relativity*, (Chicago:The University of Chicago Press) chapter 13.

19) R.P. Feynman [1949], "The theory of positrons", Phys. Rev., 76,749.

Appendix 1

This appendix contains Fortran routines which were run on a VAX computer. They sprinkle a region of 1+1 Minkowski spacetime upon which a collection of scalar fields are defined. This gives scalar fields defined on a causal set. The inverse Green's function action is calculated for these fields. Several runs are performed and statistics are calculated.

The routines RAN1, RAN3, LUDCMP, and LUBKSB are taken from the book Numerical Recipes[5].

```
      PROGRAM MAIN
      implicit double precision(a-h,o-z)
      parameter nfncs=12
      parameter bad=[talk to rafael about value]
      character*30 fnstring(nfncs)
      dimension av(nfncs),sq(nfncs)
      data fnstring/'x','t','u=x+t','v=x-t','[(u-1)(u+1)(v-1)(v+1)]^2',
     +  '(1+COSpiu)(1+COSpiv)','x[(u-1)(u+1)(v-1)(v+1)]^2',
     +  't[(u-1)(u+1)(v-1)(v+1)]^2','x(1+COSpiu)(1+COSpiv)',
     +  't(1+COSpiu)(1+COSpiv)','(1+COS5piu)(1+COS5piv)',
     +  '(1-u)[(u-1)(u+1)(v-1)(v+1)]^2'/
      istop=300
      n=0
10    do j=1,nfncs
15        x=pobox(i,j,tax)
          print *,' RUN #',i
          print *,' FUNCTION ',fnstring(j)
          if ((tax.gt.bad).or.(tax.lt.-bad)) then
              print *,' ill conditioned matrix'
              print *,' skipped over'
              goto 10
          end if
          n=n+1
          if (n.eq.1) goto 20
          sq(j)=sq(j)*(n-2)
          sq(j)= sq(j) + (av(j)**2)*(n-1)
20        av(j)=((n-1)*av(j)+x)/n
          if (n.eq.1) goto 30
          sq(j)=sq(j)-n*(av(j)**2)
          sq(j)=sq(j)+x**2
          sq(j)=sq(j)/(n-1)
30        print *,' action =',x
          print *,n,' data points give:'
          print *,' running average=',av(j)
          print *,' running standard deviation=',sq(j)**.5
          print *,' running standard error of the mean=',(sq(j)/n)**.5
```

```
40          print *
          end do
          if (n.lt.istop) goto 10
          end
```

```
      DOUBLE PRECISION FUNCTION POBOX(little_dumb,jfn,tax)
c     ---------------------------
c     nexp := expectation value of number of points to be sprinkled
c     fr := 2 x diagonal value of λδ + C (i.e. fr = 2λ)
c     ndim := number of dimensions, 2 or 4
c     ---------------------------
      parameter nfncs=12
      parameter nexp=200
      parameter mex=nexp+3*INT(SQRT(nexp))
      parameter ndim=2
      parameter fr=.5
      dimension l(mex,mex)
      double precision f(mex,nfncs),c(mex,4),cin(mex,mex)
      double precision dummy(mex,mex),indx(mex)
      dimension icheck(mex,mex)
      double precision pi,tax,sum,s,action(nfncs)
c     dimension lilac(42)
c     dimension lilac(2)
      character*3 meth
      common pi
c     ---------------------------
c     meth='sym' gives symmetric
c          green's function
c     meth='ret' gives retarded
c          green's function
c     ---------------------------
c     meth='sym'
      meth='ret'
c     data lilac/111222,111333/
c     data lilac/111222,111333,111444,111555,111666,111777,111888,
c    +    111999,222111,222333,222444,222555,222666,222777,222888,
c    +    222999,333111,333222,333444,333555,333666,333777,333888,
c    +    333999,444111,444222,444333,444555,444666,444777,444888,
c    +    444999,555111,555222,555333,555444,555666,555777,555888,
c    +    555999,314159,271828/
      if (jfn.gt.1) goto 200
```

```
      pi=2*dasin(1.0d0)
      iseed=itime(1)
c     ****************************************
c     to make portable, the call for time in
c     ITIME must be modified to correct form
c     for specific machine being used
c     ****************************************
c     iseed=456657456
c     -------------------------
c     the array lilac can be used to provide
c     prespecified seeds for ran3
c     -------------------------
c     iseed=lilac(little_dumb)
      print *,' RUN #',little_dumb
      print *,' iseed=',iseed
      x=ran3(-iseed)
10    n=npt(nexp,iseed)
c     n=6
      if (n.gt.mex) goto 10
      if (little_dumb.eq.1) goto 111
      do i=1,n
         do j=i,n
            cin(j,i)=0.
            dummy(j,i)=0.
            icheck(j,i)=0
            l(j,i)=0
            cin(i,j)=0.
            dummy(i,j)=0.
            icheck(i,j)=0
            l(i,j)=0
         end do
      end do
c     ********************************************
c     This allows interactive control over
c       # of points and iseed
c     print *,' input # of points:'
c     read *,n
c     print *,' type in random seed(6 digits or less):'
```

```
c       read *,iseed
c       ***********************************************
111      print *,' number of points =',n
        call sprinkle(mex,n,ndim,c,f,iseed,nfncs)
c       print *,' coordinates :'
c       print *,((c(i,j),j=1,ndim),i=1,n)
c       print *,' field values :'
c       print *,(f(i,9),i=1,n)
        call setc(mex,n,c,ndim,l)
c       print *,' causal matrix :'
c       print *,((l(i,j),j=1,n),i=1,n)
        call relab(mex,n,l,f,nfncs)
c       print *,' relabelled causal matrix :'
c       print *,((l(i,j),j=1,n),i=1,n)
c       print *,' relabelled field values :'
c       print *,(f(i,9),i=1,n)
        if (ndim.eq.4) call link(mex,l,n)
        if (meth.eq.'ret') call gret(mex,n,l,dummy,icheck,fr)
        if (meth.eq.'sym') then
            call gsym(mex,n,l,dummy,icheck)
            call discard1(mex,n,dummy,icheck,f,nfncs)
            call discard2(mex,n,dummy,icheck,f,nfncs)
            print *,' number of points (after discards) =',n
        end if
c       print *,' matrix to be inverted :'
c       print *,((dummy(i,j),j=1,n),i=1,n)
c       print *,((icheck(i,j),j=1,n),i=1,n)
        call inv(mex,n,dummy,cin,indx)
c       print *,' inverse of causal matrix :'
c       print *,((cin(i,j),j=1,n),i=1,n)
        tax=0.0
        if (meth.eq.'sym') then
            do i=1,n
                do j=1,n
                    sum=0.0
                    do k=1,n
                        if (i.eq.k) then
                            sum=sum+fr*cin(k,j)
```

```
                        goto 13
                    end if
                    sum=sum+icheck(i,k)*cin(k,j)
13                end do
c               print *,' identity matrix elements :'
c               print *,i,j,sum
                  if ((i.eq.j).and.(dabs(sum-1.).gt.tax)) tax=dabs(sum-1.)
                  if ((i.ne.j).and.(dabs(sum).gt.tax)) tax=dabs(sum)
                end do
            end do
        end if
        do j=1,nfncs
            s=act(mex,n,cin,f,j,nfncs)
            if (meth.eq.'sym') s=4*s
            if (meth.eq.'ret') s=2*s
            action(j)=s
        end do
200     pobox=action(jfn)
        end
```

```
      FUNCTION ITIME(n)
c     ------------------------
c     this routine generates a seed for ran3 based on the
c     time of day. it works on a VAX and must be alterred
c     to work on other machines.
c     ------------------------
      character *23 date
      character a,b,c,d,e,f
      call lib$date_time(date)
      a=date(14:14)
      b=date(17:17)
      c=date(19:19)
      d=date(20:20)
      e=date(22:22)
      f=date(23:23)
      if ((ichar(e).eq.0).and.(ichar(f).eq.0)) goto 10
      itime=(ichar(e)-48)*10000+(ichar(d)-48)*1000
     +  itime=itime+(ichar(c)-48)*100
      itime=itime+(ichar(b)-48)*10+ichar(a)-48
      itime=itime+(ichar(f)-48)*100000
      x=ran3(-itime)
10    x=ran3(1)
      if (x.lt.0.1) goto 10
      itime=int(x*1000000.)
      end
```

```
      FUNCTION NPT(nexp,iseed)
c     -------------------------
c     this routine calculates a random
c     number according to a poisson distribution
c     with expectation value nexp
c     -------------------------
      x=ran3(iseed)
      n=nexp
      p=0
      j=0
100   p=p+pdv(j,n)
      if (x.lt.p) goto 200
      j=j+1
      goto 100
200   npt=j
      end

      FUNCTION PDV(i,n)
      f=0
      do j=1,i
         f=f+alog(real(j))
      end do
      pdv=i*alog(real(n))-f-real(n)
      pdv=exp(pdv)
      end
```

```
      SUBROUTINE SPRINKLE(mex,n,ndim,coor,field,iseed,nf)
c     ------------------------
c     this routine sprinkles points into an
c     Alexandrov neighborhood of 1+1 (when ndim=2) or
c     3+1 (when ndim=4) Minkowski space
c     ------------------------
      double precision pi,x1,x2,x3,x4,x,t,coor(mex,4),field(mex,nf)
      common pi
      print *,' at start of routine sprinkle iseed is',iseed
      iseed=-iseed
      do i=1,n
          x1=ran3(iseed)
          x2=ran3(iseed)
          if (ndim.eq.4) then
              x3=ran3(iseed)
              x4=ran3(iseed)
          end if
          if (ndim.eq.2) then
              coor(i,1)=x1-x2
              coor(i,2)=x1+x2-1
          end if
          if (ndim.eq.4) then
              coor(i,4)=2*x4-1
              coor(i,1)=x1**(1./3)*coor(i,4)
              coor(i,2)=dacos(1-2*x2)
              coor(i,3)=2*pi*x3
          end if
          x=coor(i,1)
          t=coor(i,ndim)
          do j=1,nf
              field(i,j)=fld(x,t,j,nf)
          end do
      end do
      end
```

```
DOUBLE PRECISION FUNCTION FLD(x,t,j,nfncs)
double precision x,t,pi,u,v
common pi
u=x+t
v=x-t
if (j.eq.1) fld=x
if (j.eq.2) fld=t
if (j.eq.3) fld=u
if (j.eq.4) fld=v
if (j.eq.5) then
     fld=(u-1)*(u+1)*(v-1)*(v+1)
     fld=fld**2
end if
if (j.eq.6) fld=(1+dcos(pi*u))*(1+dcos(pi*v))
if (j.eq.7) then
     fld=(u-1)*(u+1)*(v-1)*(v+1)
     fld=fld**2
     fld=x*fld
end if
if (j.eq.8) then
     fld=(u-1)*(u+1)*(v-1)*(v+1)
     fld=fld**2
     fld=t*fld
end if
if (j.eq.9) fld=x*(1+dcos(pi*u))*(1+dcos(pi*v))
if (j.eq.10) fld=t*(1+dcos(pi*u))*(1+dcos(pi*v))
if (j.eq.11) fld=(1+dcos(5.*pi*u))*(1+dcos(5.*pi*v))
if (j.eq.12) then
     fld=(u-1)*(u+1)*(v-1)*(v+1)
     fld=fld**2
     fld=(1-u)*fld
end if
if ((j.lt.1).or.(j.gt.nfncs)) then
     print *,' function # out of range'
     call halt
end if
```

```
end
```

```
      FUNCTION RAN3(idum)
c     initially call with a negative integer
      implicit real*4(m)
      parameter mbig=4000000.,mseed=1618033.,mz=0.,fac=1./mbig
      dimension ma(55)
      data iff /0/
      if ((idum.lt.0).or.(iff.eq.0)) then
          iff=1
          mj=mseed-iabs(idum)
          mj=mod(mj,mbig)
          ma(55)=mj
          mk=1
          do i=1,54
              ii=mod(21*i,55)
              ma(ii)=mk
              mk=mj-mk
              if (mk.lt.mz) mk=mk+mbig
              mj=ma(ii)
          end do
          do k=1,4
              do i=1,55
                  ma(i)=ma(i)-ma(1+mod(i+30,55))
                  if (ma(i).lt.mz) ma(i)=ma(i)+mbig
              end do
          end do
          inext=0
          inextp=31
          idum=1
      end if
      inext=inext+1
      if (inext.eq.56) inext=1
      inextp=inextp+1
      if (inextp.eq.56) inextp=1
      mj=ma(inext)-ma(inextp)
      if (mj.lt.mz) mj=mj+mbig
      ma(inext)=mj
```

```
ran3=mj*fac
return
end
```

```
      SUBROUTINE SETC(mex,n,c,ndim,l)
c     -------------------------
c     this routine sets the causal matrix
c     -------------------------
      dimension l(mex,mex)
      double precision c(mex,4)
      double precision dx,dy,dz,dt,ds2
      if (ndim.eq.4) then
      do i=1,n-1
          do j=i+1,n
              dx=c(i,1)*dsin(c(i,2))*dcos(c(i,3))
              dx=dx-c(j,1)*dsin(c(j,2))*dcos(c(j,3))
              dy=c(i,1)*dsin(c(i,2))*dsin(c(i,3))
              dy=dy-c(j,1)*dsin(c(j,2))*dsin(c(j,3))
              dz=c(i,1)*dcos(c(i,2))-c(j,1)*dcos(c(j,2))
              dt=c(i,4)-c(j,4)
              ds2=dx**2+dy**2+dz**2-dt**2
              if (ds2.gt.0) goto 100
              if (dt.gt.0) l(j,i)=1
              if (dt.lt.0) l(i,j)=1
100       end do
      end do
      end if
      if (ndim.eq.2) then
      do i=1,n-1
          do j=i+1,n
              dx=c(i,1)-c(j,1)
              dt=c(i,2)-c(j,2)
              ds2=dx**2-dt**2
              if (ds2.gt.0) goto 200
              if (dt.gt.0) l(j,i)=1
              if (dt.lt.0) l(i,j)=1
200       end do
      end do
      end if
      end
```

```
      SUBROUTINE RELAB(mex,n,l,f,nfs)
c     ---------------------------
c     this routine relabels causal matrix
c     so that it is in upper triangular form
c     ---------------------------
      dimension l(mex,mex)
      double precision x,f(mex,nfs)
      do i=n,2,-1
50        do j=1,i-1
              if (l(i,j).eq.0) goto 100
              do k=1,n
                  m=l(i,k)
                  l(i,k)=l(j,k)
                  l(j,k)=m
              end do
              do k=1,n
                  m=l(k,i)
                  l(k,i)=l(k,j)
                  l(k,j)=m
              end do
              do k=1,nfs
                  x=f(i,k)
                  f(i,k)=f(j,k)
                  f(j,k)=x
              end do
              goto 50
100       end do
      end do
      end
```

```
      SUBROUTINE GSYM(mex,n,l,dummy,icheck)
c     -------------------------
c     this routine sets the symmetric
c     green's function matrix
c     -------------------------
      dimension l(mex,mex),icheck(mex,mex)
      double precision dummy(mex,mex)
      diag=0.0
      idiag=0
      do j=2,n
          do i=1,j-1
              dummy(i,j)=l(i,j)
              dummy(j,i)=l(i,j)
              icheck(i,j)=l(i,j)
              icheck(j,i)=l(i,j)
          end do
          dummy(j,j)=diag
          icheck(j,j)=idiag
      end do
      dummy(1,1)=diag
      icheck(1,1)=idiag
      end
```

```
      SUBROUTINE GRET(mex,n,l,dummy,icheck,fr)
c     ---------------------------
c     this routine sets the retarded
c     green's function matrix
c     ---------------------------
      dimension l(mex,mex),icheck(mex,mex)
      double precision dummy(mex,mex)
      do j=2,n
          do i=1,j-1
              dummy(i,j)=l(i,j)
              dummy(j,i)=0.0
              icheck(i,j)=l(i,j)
              icheck(j,i)=0
          end do
          dummy(j,j)=fr
      end do
      dummy(1,1)=fr
      end
```

```
      SUBROUTINE DISCARD1(mex,n,du,ich,f,nfncs)
c     -------------------------
c     this routine locates any rows of zeros which occur
c     in the green's function matrix and cals the routine del
c     -------------------------
      dimension ich(mex,mex)
      double precision du(mex,mex),f(mex,nfncs)
      if (n.eq.1) then
          print *,' matrix of rank 1 and so no deletion possible'
          print *,' exiting discard1 immediately'
          goto 40
      end if
      if (n.lt.1) then
          print *,' matrix of rank < 1 not possible !'
          call halt
      end if
      i=0
10    i=i+1
      j=0
20    j=j+1
      if (ich(i,j).eq.1) goto 30
      if (j.lt.n) goto 20
      call del(mex,n,ich,du,i,f,nfncs)
      print *,' a row of zeros has been deleted'
      i=i-1
30    if (i.lt.n) goto 10
40    end


  SUBROUTINE HALT
      print *,' execution of program halted'
      stop
      end
```

```
      SUBROUTINE DISCARD2(mex,n,du,ich,f,nfncs)
c     -------------------------
c     this routine locates duplicate rows from the green's
c     function matrix and calls the routine del
c
c     -------------------------
      dimension ich(mex,mex)
      double precision du(mex,mex),f(mex,nfncs)
      if (n.eq.1) then
          print *,' matrix of rank 1 and so no deletion possible'
          print *,' exiting discard 2 immediately'
          goto 50
      end if
      if (n.lt.1) then
          print *,' matrix of rank < 1 not possible !'
          call halt
      end if
      i=0
10    i=i+1
      j=i
20    j=j+1
      k=0
30    k=k+1
      if (ich(i,k).ne.ich(j,k)) goto 40
      if (k.lt.n) goto 30
      call del(mex,n,ich,du,j,f,nfncs)
      print *,' non-Hegelian point has been removed'
      j=j-1
40    if (j.lt.n) goto 20
      if (i.lt.n-1) goto 10
50    end
```

```
      SUBROUTINE DEL(mex,n,ich,du,i,f,nfncs)
c     -------------------------
c     this routine deletes the specified row and column
c     of the green's function matrix and the associated
c     element of the array of field values
c     -------------------------
      dimension ich(mex,mex)
      double precision du(mex,mex),f(mex,nfncs)
      if (n.le.1) then
          print *,' matrix of rank < 2 and so will not delete'
          call halt
      end if
      if ((i.lt.1).or.(i.gt.n)) then
          print *,' trying to delete point which does not exist !'
          call halt
      end if
      if (i.eq.n) goto 10
          do k=i+1,n
              do m=1,n
                  du(k-1,m)=du(k,m)
                  ich(k-1,m)=ich(k,m)
              end do
              do jc=1,nfncs
                  f(k-1,jc)=f(k,jc)
              end do
          end do
          do m=1,n-1
              do k=i+1,n
                  du(m,k-1)=du(m,k)
                  ich(m,k-1)=ich(m,k)
              end do
          end do
10        n=n-1
      end
```

```
      SUBROUTINE INV(np,n,a,y,indx)
c     ------------------------
c     this routine is a bridge between
c     pobox and the inversion routines
c     ------------------------
      dimension indx(np)
      double precision a(np,np),y(np,np)
      do i=1,n
          do j=1,n
              y(j,i)=0.
          end do
          y(i,i)=1.
      end do
      call ludcmp(a,n,np,indx,d)
      do j=1,n
          call lubksb(a,n,np,indx,y(1,j))
      end do
      end
```

```
SUBROUTINE LUDCMP(a,n,np,indx,d)
parameter nmax=1000
parameter tiny=1.0e-10
dimension indx(n)
double precision aamax,sum,dum,a(np,np),vv(nmax)
d=1.
do i=1,n
    aamax=0.
    do j=1,n
        if (dabs(a(i,j)).gt.aamax) aamax=dabs(a(i,j))
    end do
    if (aamax.eq.0) pause 'Singular matrix'
    vv(i)=1./aamax
end do
do j=1,n
    if (j.gt.1) then
        do i=1,j-1
            sum=a(i,j)
            if (i.gt.1) then
                do k=1,i-1
                    sum=sum-a(i,k)*a(k,j)
                end do
                a(i,j)=sum
            end if
        end do
    end if
    aamax=0.
    do i=j,n
        sum=a(i,j)
        if (j.gt.1) then
            do k=1,j-1
                sum=sum-a(i,k)*a(k,j)
            end do
            a(i,j)=sum
        end if
        dum=vv(i)*dabs(sum)
```

```
            if (dum.ge.aamax) then
                imax=i
                aamax=dum
            end if
        end do
        if (j.ne.imax) then
            do k=1,n
                dum=a(imax,k)
                a(imax,k)=a(j,k)
                a(j,k)=dum
            end do
            d=-d
            vv(imax)=vv(j)
        end if
        indx(j)=imax
        if (j.ne.n) then
            if (a(j,j).eq.0.) a(j,j)=tiny
            dum=1./a(j,j)
            do i=j+1,n
                a(i,j)=a(i,j)*dum
            end do
        end if
    end do
    if (a(n,n).eq.0.) a(n,n)=tiny
    end
```

```
      SUBROUTINE LUBKSB(a,n,np,indx,b)
      dimension indx(n)
      double precision sum,a(np,np),b(n)
      ii=0
      do i=1,n
           ll=indx(i)
           sum=b(ll)
           b(ll)=b(i)
           if (ii.ne.0) then
                do j=ii,i-1
                     sum=sum-a(i,j)*b(j)
                end do
           else if (sum.ne.0.) then
                ii=i
           end if
           b(i)=sum
      end do
      do i=n,1,-1
           sum=b(i)
           if (i.lt.n) then
                do j=i+1,n
                     sum=sum-a(i,j)*b(j)
                end do
           end if
c          print *,sum,a(i,i)
           b(i)=sum/a(i,i)
      end do
      end
```

```
      DOUBLE PRECISION FUNCTION ACT(mex,n,c,f,jl,nfncs)
c     -------------------------
c     this routine calculates the action
c     up to a constant factor
c     -------------------------
      double precision c(mex,mex),f(mex,nfncs),pi,sum
      common pi
      sum=0
      do i=1,n
          do j=1,n
              sum=sum+f(i,jl)*c(j,i)*f(j,jl)
          end do
      end do
      act=sum
      end
```

```
      SUBROUTINE LINK(mex,k,num)
c     -------------------------
c     this routine takes an input causal matrix
c     and replaces it with its link matrix
c     -------------------------
      dimension k(mex,mex),kh(1024,1024)
      do i=1,num-1
          do j=i+1,num
              kh(i,j)=k(i,j)
          end do
      end do
      do i=1,num-1
          do j=i+1,num
              do l=1,num
                  if ((k(i,l).eq.1).and.(k(l,j).eq.1)) goto 10
              end do
              goto 20
10            k(i,j)=0
20        end do
      end do
      do i=1,num-1
          do j=i+1,num
              k(i,j)=kh(i,j)
          end do
      end do
      end
```

Appendix 2

This appendix contains Fortran routines which were run on a VAX computer. They enumerate the connected partial ordered sets definable on N elements for N$\leq$ 7 (larger N's take excessive time to run). Other related data are also collected, as described in Part IV.

```
      PROGRAM GENPOS
      parameter npts=7
      parameter newfils=npts-6
      parameter ninvrows=4*(1-npts/7)+(npts/7)*(4+3*newfils)
      parameter linksize=((npts**2-(1-(-1)**npts)/2)/4)+1
      parameter nspaces=npts*(npts-1)/2
      character causal(npts,npts),lmat(npts,npts),lorg(npts,npts)
      character ch(npts,npts),ch2(npts,npts)
      character ch3(npts,npts),ch4(npts,npts),ch5(npts,npts)
      character link2(npts,npts),causal2(npts,npts)
      character ci
      dimension nlf2(ninvrows,npts),ninv2(ninvrows,npts)
      dimension nlf(ninvrows,npts),ninv(ninvrows,npts)
      dimension nlfh(ninvrows,npts)
      dimension incl(npts,npts),nincl(npts),ipts(npts)
      dimension inclh(npts),itemp(npts)
      dimension nprel(npts-1:nspaces)
      dimension npcov(npts-1:linksize-1)
      common np,lnksz,nlinks,link(linksize)
      common /count/npt
c     ------------------------
c     calculated data are recorded in file genpos.dat
c     the file genpos.cnt holds the number of accumulated
c     causal sets as the program runs, as an indicator
c     to monitor its progress
c     ------------------------
c     open (unit=2,file='genpos.dat',status='new')
      open (unit=0,file='genpos.cnt',status='new')
      write (0,*)0
      close (0)
      npt=npts
      np=npts
c     ------------------------
c     istart := smallest number of links possible
c                   connected causal set
```

```
c       istop := largest number of links possible
c       --------------------------
        istart=npts-1
        istop=linksize-1
        lnksz=linksize
        nposet=0
        nmatcon=0
        do nlinks=istart,istop
c       do nlinks=istart,istart
c       --------------------------
c       open files within which data
c       is accululated
c       --------------------------
            do ill=1,istart
                do jill=1,npts-ill
                    moo=ill*10+jill
                    open (unit=moo,status='scratch',
     +                      form='unformatted')
c                   open (unit=moo,status='new',form='unformatted')
                end do
            end do
c       --------------------------
c       initialize link configuration
c       --------------------------
            do j=1,nlinks
                link(j)=nspaces-nlinks+j
            end do
            link(nlinks+1)=0
50          do i=1,npts
                do j=1,npts
                    causal(i,j)='0'
                    lmat(i,j)='0'
                    lorg(i,j)='0'
                    ch(i,j)='0'
                    ch2(i,j)='0'
                    ch3(i,j)='0'
                end do
            end do
```

```
                do i=1,istart
                    do j=i+1,npts
                        if (l(i,j).eq.1) then
                            lorg(i,j)='1'
                            causal(i,j)='1'
                            ch(i,j)='1'
                            ch(j,i)='1'
                            ch2(i,j)='1'
                            ch2(j,i)='1'
                        end if
                    end do
                end do
c           ------------------------
c           calculate causal matrix from link matrix
c           ------------------------
                do k=1,istart-1
                    do i=k+1,istart
                        if (causal(k,i).eq.'0') goto 100
                        do j=i+1,npts
                            if ((causal(i,j).eq.'1').and.
     +                          (causal(k,j).eq.'0')) then
                                causal(k,j)='1'
                                ch(k,j)='1'
                                ch(j,k)='1'
                                ch2(k,j)='1'
                                ch2(j,k)='1'
                            end if
                        end do
100                 end do
                end do
                do i=1,npts
                    ch(i,i)='1'
                    ch2(i,i)='1'
                end do
c           ------------------------
c           calculate link matrix from causal matrix
c           ------------------------
                do i=1,istart
```

```
                do k=i+1,npts
                    if (causal(i,k).eq.'0') goto 150
                    do j=i+1,k-1
                        if ((causal(i,j).eq.'1').and.
     +                      (causal(j,k).eq.'1')) goto 150
                    end do
                    lmat(i,k)='1'
150             end do
            end do
c       -------------------------
c       test if original 'link matrix' is valid link matrix
c       by comparing with calculated link matrix
c       -------------------------
            do i=1,istart
                do k=i+1,npts
                    if (lorg(i,k).ne.lmat(i,k)) goto 300
                end do
            end do
c       -------------------------
c       test if causal set is connected by looking for zero
c       entries in (1 + C + C^T)^(N-1)
c       -------------------------
            do k=2,istart
                do i=1,istart
                    do j=i+1,npts
                        do m=1,npts
                            if ((ch2(i,m).eq.'1').and.
     +                          (ch(m,j).eq.'1')) then
                                ch3(i,j)='1'
                                goto 160
                            end if
                        end do
                        ch3(i,j)='0'
160                 end do
                end do
                do i=1,istart
                    do j=i+1,npts
                        ch2(i,j)=ch3(i,j)
```

```
                    ch2(j,i)=ch3(i,j)
                end do
            end do
            do i=1,istart
                do j=i+1,npts
                    if (ch2(i,j).eq.'0') goto 170
                end do
            end do
            goto 180
170         end do
            goto 300
c       ---------------------------
c       counting number of connected matrices
c       ---------------------------
180         nmatcon=nmatcon+1
c       ---------------------------
c       accumulate data about the causal set
c       ---------------------------
            do i=1,istart
                isum1=0
                isum2=0
                do j=i+1,npts
                    if (lmat(i,j).eq.'1') isum1=isum1+1
                    if (causal(i,j).eq.'1') isum2=isum2+1
                end do
                nlf(1,i)=isum1
                nlf(3,i)=isum2
                nlfh(1,i)=isum1
                nlfh(3,i)=isum2
            end do
            do i=2,npts
                isum1=0
                isum2=0
                do j=1,i-1
                    if (lmat(j,i).eq.'1') isum1=isum1+1
                    if (causal(j,i).eq.'1') isum2=isum2+1
                end do
                nlf(2,i)=isum1
```

```
                nlf(4,i)=isum2
                nlfh(2,i)=isum1
                nlfh(4,i)=isum2
            end do
c           ---------------------------
c           from accumulated data generate invariant matrices
c           to be used to identify relabellings
c           ---------------------------
            do j=1,4
                do m=0,istart
                    isum1=0
                    do n=1,npts
                        if (nlf(j,n).eq.m) isum1=isum1+1
                    end do
                    ninv(j,m+1)=isum1
                end do
            end do
c           ---------------------------
c           accumulate additional data for large N's
c           ---------------------------
            if (npts.lt.7) goto 190
            do i=1,istart
                ch(i,i)='0'
                ch2(i,i)='0'
                ch4(i,i)='0'
                do j=i+1,npts
                    ch(i,j)=lorg(i,j)
                    ch(j,i)='0'
                    ch2(i,j)=lorg(i,j)
                    ch2(j,i)=lorg(i,j)
                    ch3(i,j)='0'
                    ch4(i,j)=lorg(i,j)
                    ch4(j,i)=lorg(i,j)
                end do
            end do
            ch(npts,npts)='0'
            ch2(npts,npts)='0'
            ch4(npts,npts)='0'
```

```
      do nfil=1,newfils
          invnum=2+3*nfil
          do i=1,istart
              ifor=0
              do j=i+1,npts
                  do k=i+1,j-1
                      if ((ch(i,k).eq.'1').and.
     +                     (lorg(k,j).eq.'1')) then
                          ch3(i,j)='1'
                          ifor=ifor+1
                          goto 185
                      end if
                  end do
                  ch3(i,j)='0'
185               end do
              nlf(invnum,i)=ifor
              nlfh(invnum,i)=ifor
          end do
          do j=2,npts
              ibak=0
              do i=1,j-1
                  if (ch3(i,j).eq.'1') ibak=ibak+1
              end do
              nlf(invnum+1,j)=ibak
              nlfh(invnum+1,j)=ibak
          end do
          do j=1,npts
              do i=1,npts
                  do k=1,npts
                      if ((ch2(i,k).eq.'1').and.
     +                     (ch4(k,j).eq.'1')) then
                          ch5(i,j)='1'
                          goto 186
                      end if
                  end do
                  ch5(i,j)='0'
186               end do
          end do
```

```
                do i=1,npts
                    ipath=0
                    do j=1,npts
                        if (ch5(i,j).eq.'1') ipath=ipath+1
                    end do
                    nlf(invnum+2,i)=ipath
                    nlfh(invnum+2,i)=ipath
                end do
c       ------------------------
c       generate additional invariants
c       ------------------------
                do mars=0,istart
                    icount1=0
                    icount2=0
                    icount3=0
                    do i=1,npts
                        if (nlf(invnum,i).eq.mars) then
                            icount1=icount1+1
                        end if
                        if (nlf(invnum+1,i).eq.mars) then
                            icount2=icount2+1
                        end if
                        if (nlf(invnum+2,i).eq.mars) then
                            icount3=icount3+1
                        end if
                    end do
                    ninv(invnum,mars+1)=icount1
                    ninv(invnum+1,mars+1)=icount2
                    ninv(invnum+2,mars+1)=icount3
                end do
                if (nfil.eq.newfils) goto 188
                do i=1,istart
                    do j=i+1,npts
                        ch(i,j)=ch3(i,j)
                    end do
                end do
                do j=1,npts
                    do i=1,npts
```

```
                        ch4(i,j)=ch5(i,j)
                    end do
                end do
188         end do
c       ---------------------------
c       moo := hashing parameter based on number of
c       minimal and maximal elements
c       ---------------------------
            moo=ninv(1,1)*10+ninv(2,1)
            rewind moo
            icount=0
c       ---------------------------
c       test if moo is a bad value and printout data
c       for examination
c       ---------------------------
            if (moo.lt.11) then
                print *,' moo calculated to be',moo,' from this data'
                print *,'nlf:'
                do i=1,ninvrows
                    print *,(nlfh(i,j),j=1,npts)
                end do
                print *,'ninv:'
                do i=1,ninvrows
                    print *,(ninv(i,j),j=1,npts)
                end do
                print *,' Poset #',nposet,' : ',
     +              (lorg(1,j),j=1,npts),' ',
     +              (causal(1,j),j=1,npts)
                do i=2,npts
                    print *,' ',
     +                  (lorg(i,j),j=1,npts),' ',(causal(i,j),j=1,npts)
                end do
                print *
                stop
            end if
c       ---------------------------
c       compare causal set with previously retained sets
c       ---------------------------
```

```
192         read(moo,end=1310)link2,causal2,nlf2,ninv2
            icount=icount+1
            do j=1,npts
                do i=1,ninvrows
                    if (ninv(i,j).ne.ninv2(i,j)) goto 192
                end do
            end do
c           -------------------------
c           test possibility of causal set being a relabelling
c           of previous set by reordering accumulated data
c           to try and match
c           -------------------------
190         do i=1,npts
                do j=1,ninvrows
                    if (nlf2(j,i).ne.nlf(j,i)) goto 1000
                end do
                goto 1200
1000            if (i.eq.npts) goto 192
                do j=i+1,npts
                    do k=1,ninvrows
                        if (nlf2(k,i).ne.nlf(k,j)) goto 1100
                    end do
                    do k=1,ninvrows
                        mice=nlf(k,i)
                        nlf(k,i)=nlf(k,j)
                        nlf(k,j)=mice
                    end do
                    do k=1,npts
                        ci=lmat(k,i)
                        lmat(k,i)=lmat(k,j)
                        lmat(k,j)=ci
                    end do
                    do k=1,npts
                        ci=lmat(i,k)
                        lmat(i,k)=lmat(j,k)
                        lmat(j,k)=ci
                    end do
                    goto 1200
```

```
1100              end do
                  goto 192
1200          end do
c         ------------------------
c         test for match of link matrices. if no match then
c         identify classes of permutable labels based on the
c         accumulated data
c         ------------------------
              do i=1,istart
                  do j=i+1,npts
                      if (lmat(i,j).ne.link2(i,j)) then
                          ncl=1
                          do m=1,npts
                              ipts(m)=m
                              nincl(m)=1
                              do n=1,npts
                                  incl(m,n)=0
                              end do
                          end do
                          nleft=npts
1220                      n=2
1230                      if (n.gt.nleft) goto 1250
                          do k=1,ninvrows
                              if (nlf(k,ipts(1)).ne.
     +                             nlf(k,ipts(n))) then
                                  n=n+1
                                  goto 1230
                              end if
                          end do
                          nincl(ncl)=nincl(ncl)+1
                          incl(ncl,1)=ipts(1)
                          incl(ncl,nincl(ncl))=ipts(n)
                          nleft=nleft-1
                          if (n.gt.nleft) goto 1250
                          do k=n,nleft
                              ipts(k)=ipts(k+1)
                          end do
                          goto 1230
```

```
1250                    if (nleft.le.2) goto 2000
                        nleft=nleft-1
                        do k=1,nleft
                            ipts(k)=ipts(k+1)
                        end do
                        if (nincl(ncl).eq.1) goto 1220
                        ncl=ncl+1
                        goto 1220
2000                    icl=1
c       ------------------------
c       test permutations of labels and test against
c       previously retained matrix
c       ------------------------
2100                    lulu=0
                        do m=1,nincl(icl)
                            inclh(m)=incl(icl,m)
                        end do
                        call nextperm(icl,nincl(icl),incl,lulu,
     +                                          npts)
                        if (lulu.eq.1) goto 2200
                        do m=1,nincl(icl)-1
                            if (incl(icl,m).eq.inclh(m))
     +                              goto 2120
                            do n=m+1,nincl(icl)
                                if (incl(icl,m).ne.
     +                                  inclh(n)) goto 2110
                                do k=1,npts
                                    ci=lmat(inclh(m),k)
                                    lmat(inclh(m),k)=
     +                                      lmat(inclh(n),k)
                                    lmat(inclh(n),k)=ci
                                end do
                                do k=1,npts
                                    ci=lmat(k,inclh(m))
                                    lmat(k,inclh(m))=
     +                                  lmat(k,inclh(n))
                                    lmat(k,inclh(n))=ci
                                end do
```

```
                                    mice=inclh(m)
                                    inclh(m)=inclh(n)
                                    inclh(n)=mice
                                    goto 2120
2110                                end do
2120                           end do
                               do m=1,istart
                                    do n=m+1,npts
                                         if (lmat(m,n).ne.
     +                                    link2(m,n)) goto 2000
                                    end do
                               end do
                               goto 300
2200                           if (icl.eq.ncl) goto 192
                               do m=1,nincl(icl)
                                    itemp(m)=incl(icl,m)
                               end do
                               do m=1,nincl(icl)
                                    incl(icl,m)=itemp(nincl(icl)+1-m)
                               end do
                               icl=icl+1
                               goto 2100
                          end if
                     end do
                end do
                goto 300
c       -------------------------
c       after surviving uniqueness tests,
c       save data and accumulate counts
c       -------------------------
1310        write(moo)lorg,causal,nlfh,ninv
            nposet=nposet+1
            nrel=0
            do i=1,istart
                do j=i+1,npts
                    if (causal(i,j).eq.'1') nrel=nrel+1
                end do
            end do
```

```
            nprel(nrel)=nprel(nrel)+1
            npcov(nlinks)=npcov(nlinks)+1
c           write (unit=2,fmt=*)'—nlinks=',nlinks,' npcov(nlinks)=',
c     +           npcov(nlinks),' —-nrels=',nrel
c*********************************************************
c     want now to write the number of posets so that if
c     program crashes, can know how many were collected
c     to that point
                open (unit=0,file='genpos.cnt',status='old')
                rewind 0
                write (0,*)nposet
                close (0)
c*********************************************************
c           write (unit=2,fmt=*)'nlf:'
c           do i=1,ninvrows
c           write (unit=2,fmt=*)(nlfh(i,j),j=1,npts)
c           end do
c           write (unit=2,fmt=*)'ninv:'
c           do i=1,ninvrows
c           write (unit=2,fmt=*)(ninv(i,j),j=1,npts)
c           end do
c           write (unit=2,fmt=*)' Poset #',nposet,' : ',
c     +           (lorg(1,j),j=1,npts),' ',
c     +           (causal(1,j),j=1,npts)
c           do i=2,npts
c           write (unit=2,fmt=*)' ',
c     +           (lorg(i,j),j=1,npts),' ',(causal(i,j),j=1,npts)
c           end do
c           write (unit=2,fmt=*)
300         lulu=0
c     ------------------------
c     get next link configuration
c     ------------------------
            call nextlink(lulu)
            if (lulu.eq.0) goto 50
c     ------------------------
c     after finishing all link configurations,
c     throw away all data so can start fresh
```

```
c       -------------------------
            do ill=1,istart
                do jill=1,npts-ill
                    moo=ill*10+jill
                    close (moo)
                end do
            end do
        end do
c       -------------------------
c       printout all data
c       -------------------------
        print *,' for',npts,' elements there are',nposet,' posets'
                open (unit=0,file='genpos.cnt',status='old')
                rewind 0
                write (0,*)' for',npts,' points there are',
     +                  nposet,' posets'
                write (0,*)' there are',nmatcon,' matrices which',
     +                  ' describe connected posets'
                write (0,*)
                write (0,*)' # rel.s # connected posets'
                write (0,*)'-------------------------'
                do jj=istart,nspaces
                    write (0,*)jj,nprel(jj)
                end do
                write (0,*)
                write (0,*)' # links # connected posets'
                write (0,*)'-------------------------'
                do jj=istart,istop
                    write (0,*)jj,npcov(jj)
                end do
                close (0)
        end



        SUBROUTINE NEXTLINK(lulu)
c       -------------------------
c       this routine changes the link configuration
c       -------------------------
```

```
      common npts,linksize,n,l(100)
      i=0
10    i=i+1
      if (l(i).eq.i) goto 10
      if (i.eq.n+1) goto 20
      l(i)=l(i)-1
      if (i.eq.1) goto 30
      do j=i-1,1,-1
          l(j)=l(j+1)-1
      end do
      goto 30
20    lulu=1
30    end


      FUNCTION L(i,j)
c     -------------------------
c     this routine calculates the value of link matrix
c     element based upon link configuration
c     -------------------------
      common npts,linksize,nlinks,link(100)
      n=isum(i,j)
      do k=1,nlinks
          if (link(k).eq.n) then
              l=1
              goto 100
          end if
      end do
      l=0
100   end

      FUNCTION ISUM(i,j)
      common /count/npt
      isum=0
      do k=1,npt-i
          isum=isum+k
      end do
      isum=isum+i-j+1
```

```
      end


      SUBROUTINE NEXTPERM(icl,nincl,incl,lulu,npts)
c     ------------------------
c     this routine performs a permutation
c     of labels within a permutation class
c     ------------------------
      dimension incl(npts,npts)
      dimension lh(20)
      common np
      lpos=1
      nh=1
      do i=1,npts
          lh(i)=0
      end do
10    if (incl(icl,lpos).lt.incl(icl,lpos+1)) goto 20
      if (lpos.eq.npts-1) then
          lulu=1
          goto 40
      end if
      lpos=lpos+1
      goto 10
20    do i=1,lpos
          if ((incl(icl,i).lt.incl(icl,lpos+1)).and.(incl(icl,i).gt.nh))
     +     nh=incl(icl,i)
      end do
      nh2=incl(icl,lpos+1)
      incl(icl,lpos+1)=nh
      do i=1,lpos
          if (incl(icl,i).eq.nh) then
              incl(icl,i)=nh2
              goto 25
          end if
      end do
25    do i=1,lpos
          m=incl(icl,i)
          lh(m)=m
```

```
        end do
        i=1
        j=1
30      if (lh(j).eq.0) then
            j=j+1
            goto 30
        end if
        incl(icl,i)=lh(j)
        if (i.eq.lpos) goto 40
        i=i+1
        j=j+1
        goto 30
40      end
```

CURRICULUM

Name: Alan Ronald Daughton

Born: March 24, 1963

Nationality: United States

Education:

Ph.D. in Physics from Syracuse University, Syracuse, NY, U.S.A., 1993.
M.S. in Physics from Syracuse University, Syracuse, NY, U.S.A., 1988.
B.S. in Physics, from California Institute of Technology, Pasadena, CA, U.S.A., 1986.