Toward an Information Mechanics

Michael Manthey
Department of Mathematics & Computer Science
Aalborg University
Frederik Bajersvej 7
9220 Aalborg Ø DENMARK

Abstract

This paper presents a chain of reasoning that makes an information mechanics *a plausible goal. A radically new model of distributed computation that exceeds Turing's sequential model refutes the perception that quantum mechanics cannot be captured computationally (§2,3). Our new model, called the* phase web *paradigm, is itself captured naturally by a physically relevant mathematics, that of a Clifford algebra (§4). The basic features of the computational model are shown to have natural counterparts in current physical theory (§5), and we close with a discussion of the implications of the framework presented for the fabrication of nano-scale hardware (§6).*

1 Introduction

Although the generality of the word 'information' makes it difficult to define, it is at least clear that to produce a theory of its form and transformation is to produce a theory of everything. The present paper does not present such a theory! Rather, it presents a chain of reasoning that makes such an information-based theory appear possible.

The origins of the ideas presented here lie, perhaps surprisingly, in the author's search for a simultaneous solution to the classical and otherwise unsolved AI problems of Learning and Planning. The latter is roughly the problem of making a computer react 'intelligently' to impulses from its environment; the former is, equally roughly, inferring knowledge from experience. In the present context, these problems equate to the scientific investigation of Nature (Learning) and the production of a theory of possible actions and their effects (Planning). Given that we seek a *computational* model of Nature, it follows that this model will be a mechanistic one, but where the mechanism in question is based on the the acquistion and manipulation of *information*, not materia, and hence ought to be called *neo*-mechanistic.

The first step in producing such an *information mechanics* is to refute the currently widespread view that computation in principle cannot capture quantum mechanical phenomena. This is the topic of the next two sections. The second of these also shows how to build a (radically) new model of distributed computation by focusing on the act of observation.

2 Turing's Box

In the 1930's, Turing presented a *model* of a device - the Turing Machine (TM) - which could in principle carry out any 'effective procedure'. The latter term is formally undefined, having the same status as 'point' and 'straight line' in classical geometry. It can reasonably *be* undefined because it seems intuitively clear what one means: given starting and ending states, and operations for transforming one state to another, an 'effective procedure' will specify how to get from the start to the end.

All contemporary sequential computers can be viewed as merely (rather more) efficient versions of the TM's unbounded tape and associated read/write operations [=memory], state-encoding symbols [=bits], and finite control unit [=cpu]: the TM is simply a mathematically tractable abstraction from 'practical details'. A TM is a *sequential* computational engine and deterministic in its operation: the final state and the path of transformations followed thereto are a priori predictable. Example results of the TM's formal tractability are that it can be proven that no additional computational power accrues from attaching multiple tapes, and perhaps more surprising, that a *non*-deterministic TM is no more powerful than a deterministic one. Thus it already seems that computation's ability to model QM's well-known nondeterminism has received a mortal blow.

To make matters worse, the material and macroscopic foundation of all contemporary computers would as well seem to eliminate the possibility of capturing more exotic QM phenomena like wave-particle duality and non-locality. And finally, parallel computation as generally understood provides no escape from Turing's box, as argued clearly and closely by Penrose [Pen89].

Clearly, if there is a way for computation to escape from Turing's box, it is well hidden. Space does not permit a fully detailed exposition of how nevertheless to do this, so only the most critical aspects appear below, and the reader's willingness to be convinced is herewith invoked.

As might be expected, the place to look is in the unstated assumptions, of which we find three (which appear to be distinct, but will ultimately turn out to be aspects of the same thing). The first of these assumptions is that

0-8186-6715-X/94 $4.00

the initial abstraction away from 'practical details' has left nothing crucial behind, ie. the mechanism of the TM is well-specified.

The second unstated assumption is that

computation consists only of state ***transformations,***

which turns out to entail a third assumption about

which concept of hierarchy is (implicitly) deployed.

The latter two will be treated in the following section. As an aside, it is worth noting that any computation which does not halt, ie. reach some pre-defined final state, is termed 'not computable'. Hence any system which is designed a priori *not* to halt (for example all operating systems, computer networks, real-time systems, and living systems) is consigned to a computation-theoretic limbo. One's suspicions should thus be aroused that something is amiss.

Regarding the specification of the TM's mechanism, it is tacitly assumed that the relationship between the control unit and the tape memory is unproblematic. However, this ignores the physical reality of the *separateness* of the control unit vis a vis the memory mechanism, and hence the *coordination* of their respective activities. In short, what *exactly* is the control unit doing while the tape is being read, written, or advanced, and vice versa, what is the tape mechanism doing while the control unit is active?

The problem is that each of the two units must *wait* for the other to *signal* it to begin[1]. The tacit assumption has been that this coordination is implicit in the 'sequencing' from a control unit half-transition ("it's time to read from the tape") to a tape state-transition ("it's time to read my tape") to the final control unit half-transition ("the tape is done reading, so update my control state and start a new operation").

It is important to realize that the argument being advanced is *not* dependent on the actual memory mechanism, but rather on the implicit *physical* separation between the control and memory units. Such a separation of functionality will *always* entail the above coordination. [2]

Suppose then that we try to avoid this problem by positing that the memory is somehow integral with the control unit via some heretofore undiscovered, dynamically extendable, memory technology. The 'locus of control' will then sequence unproblematically between control and memory operations. But what *is* this mysterious 'locus of control' that knows when one operation is complete and thence that it is to begin the next? One is forced, it would seem, to posit some underlying 'interpreter' or 'micro-program' which defines this. But such a micro-program invokes the original structure, so clearly we have begun an infinite logical regression[3]. Notice, cf. the third of the aforementioned assumptions, that this regression is of a hierarchical nature.

The above reasoning, brief though it is, will hopefully have at least sown some seed of doubt. Those familiar with operating system kernels or communication protocols will recognize the basic issue: the specification of a TM leaves unmentioned all issues of synchronization. Indeed, there is no place at all in the Turing model for the basic synchronization primitives *wait* and *signal*: they have no 'value' in the usual input-output, functional sense of 'state transformation', and are in principle invisible to the computation. The thrust of the above argument is that while synchronization may be invisible, its necessity for a complete definition of the TM's mechanism is *incontrovertible*. To wave it airily aside as 'the price of abstraction' serves only to vitiate further the claim of the TM's universality.

We therefore close this section with a brief explanation of the computational mechanism called *synchronization*. The two operations *wait* and *signal* operate on an entity called a 'binary semaphore', denoted S. S contains a single bit of local state (denoted s) which can take on two mutually exclusive values, denoted 1 and $\tilde{1}$. Define now *wait* and *signal* as follows:

	S.s=1	S.s=$\tilde{1}$
wait:	S.s $\leftarrow \tilde{1}$; return	continue waiting
signal:	return	S.s $\leftarrow$ 1; return

The effect of these definitions is to ensure that a given sequential computation will stall (namely when s=$\tilde{1}$) until some other computation *signals* it (which sets s to 1). Furthermore, a successful *wait* sets s to $\tilde{1}$, thus ensuring that no other computation can follow 'on its heels'. Notice that

- no 'value' is returned by either operation. Rather, each computation simply proceeds on its way as if nothing had happened;
- no information is exchanged between *wait*ing and *signal*ling computations;
- the effect of the synchronization cannot be 'observed' locally (cf. preceding item) but will be globally visible as a correlation between events in the system as a whole [Man92];
- the overall effect is to *order* events - namely the respective *wait* and *signal* events - belonging to

[1] We discuss coordination in terms of *wait / signal*, rather than polling on shared memory; this is of no greater consequence, since coordinating over shared memory implies the need for *wait / signal*, Dekker's algorithm notwithstanding.

[2] The idea of a 'tape' memory was introduced to achieve a plausibly 'unbounded' memory (simply splice additional tape on when one starts to run out). Lacking unbounded memory, a TM becomes a finite state machine, which has considerably less abstract computational power.

[3] To invoke some notion of continuum 'limit' would seem to violate the basic discreteness we associate with computation.

two *different* computations, such that (presuming S.s=$\tilde{1}$ initially) the *wait* in the one computation will always be after the *signal* in the other. No more and no less.

The classical purpose to which semaphores are put is to achieve *mutual exclusion* between two sequential computations. Consider the following two computations P,Q which use two semaphores X,Y to exclude each other from a sensitive manipulation:

```
P                 |  Q
                  |
repeat            |  repeat
 ...              |   ...
 wait(X)          |   wait(Y)
 z := z + 1       |   f(z); z := 0
 signal(Y)        |   signal(X)
forever           |  forever
```

Assuming z=0, X.s=$\tilde{1}$ and Y.s=1 initially, process P will be blocked until process Q eventually prints and zeroes z, whereafter it *signals* P that it may proceed. Now the converse obtains: Q will block if it tries to change z while P is doing so. If we imagine that P is counting cars passing a sensor and Q is computing some function f of the intermediate values, then a little analysis will reveal that without the synchronization, counts will be lost. On the other hand, *with* the synchronization no counts will be lost regardless of the relative speeds of P and Q. This alternating pattern of control is incidentally that which obtains between a TM's control and tape units.

Notice that

- Building on the preceding item, and viewing "time" as a 1-1 mapping of the events constituting a given computation to a local time axis, we see that mutual exclusion contains the germ of sequential time. In general, each computation constitutes a local *relative* time frame, which frame obtains meaning only via synchronization with other computations' frames.

- In the example above, one can conceptualize the alternating mutual exclusion between the two computations in terms of a single 'synchronization token' - which we call a 'stick' - that is passed between them. At all times there is exactly *one* stick present, either in one of the semaphores or implicitly 'owned' by one of the computations. Such a conserved stick reflects the existence of a 'resource invariant'; [Man92] argues the interpretation of this concept as the computational equivalent of quantum number conservation laws, and uses it to explain how the EPR 'paradox' is not a paradox at all.

3 Observing the Universe

An implicit claim of the Turing model is that a single sequence of computational events can capture all essential aspects of computation, or as we put it earlier, *computation consists only of state* ***transformations***. To refute this claim, consider the following gedanken experiment:

The coin demonstration *- Act I. A man stands in front of you with both hands behind his back, whilst you have one hand extended in front of you, palm up. You see the man move one hand from behind his back and place a coin on your palm. He then removes the coin with his hand and moves it back behind his back. After a brief pause, he again moves his hand from behind his back, places what appears to be an identical coin in your palm, and removes it again in the same way. He then asks you, "How many coins do I have?".*

It is important at the outset to understand that the coins are *formally* identical: indistinguishable in every respect. If you are not happy with this, replace the coins with electrons or geometric points (or synchronization sticks!).

The indistinguishability of the coins now agreed, the most inclusive answer to the question is "One or more than one", an answer which exhausts the universe of possibilities given what you have seen, namely *at least* one coin. There being exactly two possibilities, the outcome can be encoded in one bit of information. Put slightly differently, when you learn the answer to the question, you will per force have received one bit of information.

The coin demonstration *- Act II. The man now extends his hand and you see that there are two coins in it. [The coins are of course identical.]*

You now know that there are two coins, that is, *you have received one bit of information.* We have now arrived at the final act in our little drama.

The coin demonstration *- Act III. The man now asks, "Where did that bit of information come from??"*

Indeed, where *did* it come from?! Since the coins are indistinguishable, seeing them one at a time will never yield an answer to the question[4]. Rather, *the bit originates in the simultaneous presence of the two coins.* We call such a confluence a *co-occurrence.* In that a co-occurrence, by demonstration a bona fide computational entity, is 'situational' rather than 'transformational', the second of the three TM-model assumptions is shown to be false.

At this juncture, we hasten to mention that we are dealing here with *local* simultaneity, so there is no collision with relativity theory. Indeed, Feynman [Feyn65, p.63] argues from the basic principle of relativity of motion, and thence 'Einstein locality', that if *anything* is conserved, it must be conserved *locally.* See also [Phi91].

Returning to our discussion of Turing's model, we see from the coin demonstration that there is information, *computational information,* available in the universe *which* ***in principle*** *cannot be obtained se-*

[4] An statistical analysis conducted together with my colleague Søren Højsgaard yielded the conclusions (1) a Bayesian analysis is not indicated; (2) the most intuitively appropriate probability distribution for the expected number of coins given repeated showings of coins appears to be a Poisson distribution, which distribution unfortunately relies on an absolute time axis, which axis is fundamentally irrelevant to the problem.

quentially. Thus we have in the coin demonstration a compelling argument that, at the very least, the Turing model of computation fails to capture all relevant aspects of computation: it is semantically incomplete, and the thing it ultimately lacks is *space-time* - space: co-occurrence, time: mutual exclusion. Synchronization operators represent precisely the way computations can express space-time relationships and give them semantic content.

This can be taken further. Suppose we replace the coins by synchronization sticks, which are surely indistinguishable ('inscrutable' according to [PR81]'s classification). We can then say that the information received is indicative of the fact that two states (represented by their sticks) do not mutually exclude each other. Furthermore, since (1) by the definition of simultaneity, the two states occur neither before nor after each other, and (2) sticks represent what might be called 'naked instants' of time, (3) we see that in the computational frame of the co-occurrence itself, *there is no time.* Physicists call such an entity a (Wheeler-Feynman) boson; we present additional evidence for the correctness of the identification of co-occurrences with bosons later. It is well known that photons (which are bosons) are used to construct space-time.

[To very briefly dispose of the most common counter-arguments:
Q: Whatever you do, it can be simulated on a TM.
A: You can't 'simulate' co-occurrence sequentially, cf. the coin demo.
Q: But you can only check for co-occurrence sequentially - there's always a Δt.
A: This is a technological artifact: think constructive/destructive interference, which reflects the phase relationship between two entities with just one datum.
Q: Co-occurrence is primitive in Petri nets, but these are equivalent to finite state automata.
A: We in effect postulate growing Petri nets, both in nodes and connections.]

We will now show how to build a "theory structure" (for lack of a better term) from the information obtained from co-occurrences. Consider the following 'blocks world' scenario.

The blocks demonstration. *Imagine two 'places', p and q, each of which can contain a single 'block'. Each of the places is equipped with a sensor, s_p respectively s_q, which can indicate the presence or absence of a block. The sensors are the* only *source of information about the state of the places. The two states of a given sensor s are mutually exclusive, so a place is always either 'full', denoted (arbitrarily) by s, or 'empty', denoted by $\tilde{s}$. Suppose there is a block on p and none on q. This will allow us to observe the co-occurrence $\{s_p, \tilde{s}_q\}$. From this we learn that having a block on p does not exclude not having a block on q. Suppose at some other instant (either before or after the preceding) we observe the opposite, namely $\{\tilde{s}_p, s_q\}$. We now learn that not having a block on p does not exclude having a block on q. What can we conclude?*

First, it is important to realize that although the story is built around the co-occurrences $\{s_p, \tilde{s}_q\}$ and $\{\tilde{s}_p, s_q\}$, everything we say below applies equally to the 'dual' pair of co-occurrences $\{s_p, s_q\}$ and $\{\tilde{s}_p, \tilde{s}_q\}$. After all, the designation of one of a sensor's two values as '$\sim$' is entirely arbitrary. It is also important to realize that the 'places' and 'blocks' are story 'props': all we really have is two two-valued sensors reflecting otherwise unknown goings on in the surrounding environment. These sensors constitute the *boundary* between us (or some artificial entity) and this environment.

Returning to the question posed, we know that s_p excludes $\tilde{s}_p$ and similarly s_q excludes $\tilde{s}_q$. Furthermore, we have observed the co-occurrence of s_p and $\tilde{s}_q$ and vice versa. Since the respective parts of one co-occurrence exclude their counterparts in the other co-occurrence (cf. first sentence), we can conclude that the co-occurrences *as wholes* exclude each other.

Take this now a step further. The transition $s_p \rightarrow \tilde{s}_p$ is indicative of some *action* in the environment, as is the reverse, $\tilde{s}_p \rightarrow s_p$. The same applies to s_q. Perceive the transitions $s_p \leftrightarrow \tilde{s}_q$ and $s_q \leftrightarrow \tilde{s}_q$ as two sequential computations, each of whose states consists of a single value-alternating bit of information. By assumption these two computations are completely independent of each other. At the same time, the logic of the preceding paragraph allows us to infer the existence of a third computation, a third *compound* action, with the state transition $\{s_p, \tilde{s}_q\} \leftrightarrow \{\tilde{s}_p, s_q\}$. In effect, in combining in this way two single-bit computations to yield one two-bit computation, we have 'lifted' our conception of the actions performable by the environment to a new, higher, level of abstraction. This inference we call *co-exclusion*, and can be applied to co-occurrences of any arity > 1 where at least two corresponding components have changed.

Let us now briefly examine the hierarchical transition induced by the co-exclusion inference. The usual, and generally *tacitly* assumed, hierarchy relation in computing and much mathematics, is function composition, $f \circ g = f(g)$, where the idea is that the 'result' of g is to be 'composed' with, or 'put into', f. Function composition captures for example the idea of a 'subroutine' very neatly, and an entire sequential computation E can be viewed as $e_\Omega(e_{n-1}(\ldots(e_\alpha))\ldots)$. Notice however that $\circ$ implicitly imposes an *order* on the computations represented by f and g: first do g, then do f. This basic property of $\circ$ means that any consideration of co-occurrence is immediately impossible.

Furthermore, the values we are receiving from the sensors qua *values* are entirely predictable and unsurprising. That which is unpredictable and surprising is *when* they occur, and this relative to each other. Hence, in essence what we are doing is interpreting a sensor's value as a synchronization stick indicating when the sensor is in a particular state. If we were to program a computer to observe co-occurrences, apply co-exclusion to them, and model execution therewith (which we have done), the code will consist entirely of *wait* and *signal* operations. As pointed out earlier, neither of these operations can be meaningfully viewed as a 'function'.

The conclusion we draw from the preceding two paragraphs is that the hierarchical transition effected by co-exclusion cannot be described in terms of a

standard function composition hierarchy. The latter is implicitly and hopelessly sequential in its conception; the arguments to co-exclusion are co-occurrences, which cannot be captured sequentially at all; and the computational mechanism describing what's going on (synchronization) is not even functional in character.[5] Rather, instead of the function - sub-function relationship, co-exclusion captures the part-whole relationship, and this in a pure process-oriented context. We have thus established the third of our three objections to the Turing model. We will see in the following section that co-exclusion has a topological interpretation.

This temporally-based part-whole hierarchy can - via interpreting resource invariants as stick movement on closed paths - be viewed as a hyper-cyclic hierarchy. A function composition hierarchy has difficulty answering questions like, "But what is (say) a quark made out of?". In our view, the problem here is that such things as quarks are being viewed in terms of what they 'do', their 'doing-ness'. In contrast, the cycle hierarchy is founded on co-occurrence, whose timeless 'is-ness' effectively grounds such questions by referring to a clearly defined sensory boundary. In this way, the door is kept open to considering, as Leibniz intuited, that everything is ultimately defined by the presence of everything else.

Finally, with regard to practical computing, it is worth mentioning that the cycle hierarchy and co-exclusion-based actions together provide a conceptual platform for realizing distributed computations that goes *far* beyond such contemporary technologies as 'servers' and 'remote procedure call'. To demonstrate this falls however well outside the confines of the present topic. We will however very briefly sketch how actions built from co-exclusion form a computational mechanism.

Once the required pair of co-excluding co-occurrences has occurred[6], a multi-threaded action is instantiated as a new entity. One of the threads keys on the (co)-occurrence $\{s_p, s_q\}$ and a request (goal) for either $s_p \rightarrow \tilde{s}_p$ or $s_q \rightarrow \tilde{s}_q$, and the other on $\{\tilde{s}_p, \tilde{s}_q\}$ and a goal for either $\tilde{s}_p \rightarrow s_p$ or $\tilde{s}_q \rightarrow s_q$. Since these co-occurrences exclude each other, only one of these threads will be activated at a time. When these preconditions occur, the action issues requests for (say) $s_p \rightarrow \tilde{s}_p$ and $s_q \rightarrow \tilde{s}_q$. Thus a cascade of transformation goals propagates and activates other actions. Actions carried out at the boundary ('effectors') affect the environment, causing the sensors to reflect the new conditions. Etc.

Finally, given that every action possesses an innate polarity based on its orientation ($\{s_p, s_q\} \rightarrow \{\tilde{s}_p, \tilde{s}_q\}$ vs. $\{\tilde{s}_p, \tilde{s}_q\} \rightarrow \{s_p, s_q\} \mapsto \pm 1$), co-occurrences of such action polarities can themselves be subjected to the co-exclusion inference, producing a meta-level of description. Inasmuch as co-occurrence can be viewed as creating information, such meta-actions will, like the simple actions we have seen heretofore, exhibit emergent behavior. Furthermore, although it falls outside the scope of this paper to argue the point, the number of such meta-levels is in fact unbounded!

The next section of this paper analyzes the abstract properties of co-occurrences, co-exclusion, and actions, an analysis that leads to an appropriate mathematics, that of a Clifford algebra.

4 A Vector Semantics for Distributed Computations.

By way of introduction, the word 'semantics' in a computational context refers to the construction of mathematical models serving as abstractions of the symbol-level understanding of what a program will do. In particular, a semantic understanding of a program applies to all possible executions thereof. The semantics of sequential programs is completely understood, cf. the operational semantics of Plotkin, the axiomatic semantics of Hoare, and the denotational semantics of Scott & Strachey; the same can, unfortunately, not be said of concurrent systems. All computational semantic theory, including that of multi-process (ie. non-sequential) systems, follows a clear line of development from Hilbert's 14th problem to Gödel's anti-solution thereof to the work of Church and Turing, and thence to the above (1970's) semantics, up to current work. It is fair to say, though with various footnotes, that this entire line of development contains a strong overall tone of mathematical logic.

In contrast, the overall tone of the semantics of multi-process systems sketched in this section is that of physical mathematics. 'Execution' or 'state change' is viewed in terms of rotations in phase space, a view not found in traditional semantic approaches. Its use of vectors rather than scalars is untraditional, and in general the approach is radically different.

The other introductory comment refers to the vector spaces and algebra the semantics inhabits. First, we always think of the former as being discrete, and the presumption is that given the relatively modest demands we make on them, this is ultimately okay, mathematically speaking. Second, the vector spaces do *not* represent 'real' 3- or 3+1-dimensional space, nor embeddings therein, or anything of their kind. Rather, they are spaces representing distinction-coding *symbols* arising from co-exclusion, and for example the distinctions representing the left/right, up/down etc. of ordinary space-time are products of the dynamic growth of the dimensionality of the distinction-space.

In the following, we first examine the mathematical structure of co-occurrences, and thence of co-exclusion-based actions. We then show how these two fundamental entities can be viewed from a vector-space perspective, in terms of a Clifford algebra. Finally, within this framework, we provide a mathematical basis for the basic insight of inferring actions from complementary co-occurrences.

4.1 Co-occurrence.

The analysis of co-occurrence is tricky because, although the sensor-events constituting a co-occurrence are by definition *un*-ordered, there is nevertheless an

[5] See also [Hew85] for a different but related argument.

[6] The discussion also applies to the dual. Thus co-exclusion on two sensors can generate two distinct actions.

ordering present, namely that deriving from the fact that we must name the sensors uniquely in order to distinguish them from each other. A concrete version of the problem is that the sensor values will be placed in distinct storage locations, and the various possible ways this can be done must all be equivalent. Our analysis therefore proceeds by the device of considering all possible orderings in order to express co-occurrence's order-independent property.

Consider as an example the set of possible 3-co-occurrences of A, B, C, of which there will be $3! = 6$ possible order permutations. These are, explicitly,

1	a	b
(A,B,C)	(B,A,C)	(A,C,B)
(123)	(213)	(132)

c	d	e
(C,B,A)	(B,C,A)	(C,A,B)
(321)	(231)	(312)

where the middle row shows the actual permutations, the third row this same information in place-numeral form, and the first row convenient mnemonics. The numeral form can be given the following *permutation operator* interpretation:

$1 = (123)$ = do nothing
$a = (213)$ = interchange the 1st and 2nd fields
$b = (132)$ = interchange the 2nd and 3rd fields
$c = (321)$ = interchange the 1st and 3rd fields
$d = (231)$ = first do c, then do a
$e = (312)$ = first do c, then do b

For example, (123) used as an operator means "choose the first element, choose the second element, choose the third element" of the entity to which it is applied; similarly, (213) means "choose the second element, choose the first element, choose the third element". 'Multiplying' on the right, hence, $ba = (132)(213) = (312) = e$. Notice that the members of the set $\{1, a, b, c, d, e\}$ are simultaneously *things* and operations, just as the number "2" can represent both a magnitude ("two aces") and an operator ("double your bet").

The 'multiplication' table for the composition of 3-permutations is:

1	a	b	c	d	e
a	1	e	d	c	b
b	d	1	e	a	c
c	e	d	1	b	a
d	b	c	a	e	1
e	c	a	b	1	d

Since $aa = bb = cc = 1$, the elements a, b, c are their own inverses, and d, e are each other's inverse. It can also be verified that $1x = x1 = x$ and that $x(yz) = (xy)z$. Hence a 3-co-occurrence is a group, also known as the *symmetric* group on three elements, S_3; in general, S_n is a group, contains $n!$ elements, and characterizes (for example) purely spatial rotations.

Notice that by expressing the group in operator form, it makes no difference *what* it is we are permuting. This independence from some "basis" set of things to be permuted means that the properties of the group can be developed and examined independently of their application. However, a basis set of elements, for example a sensor tuple (A, B, C), can always be supplied to make a concrete computation; for example, $(A,B,C)cb = (A,B,C)(321)(132) = (A,B,C)(312) = (C,A,B)$.

We now attempt to model the Block Demonstration using the above analysis. Interpreting A, B as places, C as a 'Hand', and the position of the block as being indicated by the permuted position of the symbol 'A', the following calculation expresses the movement of a block from A to B as a sequence of purely spatial rotations: the move from A to Hand is $(A,B,C)c = (C,B,A)$, and $(C,B,A)b = (C,A,B)$ moves the block from the Hand to B. Notice however that this formulation does not express, except very implicitly, the changes which actually take place in the sensors, namely that their values invert in a very particular temporal pattern. Inversions constitute a kind of change which this group cannot express: it contains only '1', and in particular no '$\tilde{1}$'.

4.2 Co-exclusion.

We therefore move on to co-exclusion-based actions, which differ fundamentally from co-occurrences exactly in their expression of a change in the world via the inversion of a sensory input, i.e., $A \longrightarrow \tilde{A}$.

We begin with 2-ary actions. These have two components, and the operations that can occur include both permutation of places and inversion; hence in general an n-action will have $2^n n!$ elements, and a 2-action eight. These are

1	$\tilde{1}$	a	$\tilde{a}$
(A,B)	$(\tilde{A},\tilde{B})$	$(A,\tilde{B})$	$(\tilde{A},B)$
(12)	$(\tilde{1}\tilde{2})$	$(1\tilde{2})$	$(\tilde{1}2)$

b	$\tilde{b}$	c	$\tilde{c}$
$(\tilde{B},A)$	$(B,\tilde{A})$	(B,A)	$(\tilde{B},\tilde{A})$
$(\tilde{2}1)$	$(2\tilde{1})$	(21)	$(\tilde{2}\tilde{1})$

where $(1\tilde{2})$ means 'choose the first, choose the second and invert it', and we recycle the operator labels a, b, c. The table for their composition is

1	$\tilde{1}$	a	$\tilde{a}$	b	$\tilde{b}$	c	$\tilde{c}$
$\tilde{1}$	1	$\tilde{a}$	a	$\tilde{b}$	b	$\tilde{c}$	c
a	$\tilde{a}$	1	$\tilde{1}$	c	$\tilde{c}$	b	$\tilde{b}$
$\tilde{a}$	a	$\tilde{1}$	1	$\tilde{c}$	c	$\tilde{b}$	b
b	$\tilde{b}$	$\tilde{c}$	c	$\tilde{1}$	1	a	$\tilde{a}$
$\tilde{b}$	b	c	$\tilde{c}$	1	$\tilde{1}$	$\tilde{a}$	a
c	$\tilde{c}$	$\tilde{b}$	b	$\tilde{a}$	a	1	$\tilde{1}$
$\tilde{c}$	c	b	$\tilde{b}$	a	$\tilde{a}$	$\tilde{1}$	1

Suffice it to say that $\{1, \tilde{1}, a, \tilde{a}, b, \tilde{b}, c, \tilde{c}\}$ under composition form a group we will call $\mathcal{X}^2$, an *exclusion* group. Looking at the table, we can see that $aa = cc = 1$ but $bb = \tilde{1}$, and similarly for the inverses. In addition, $ab = \sim ba, bc = \sim cb$, and $ca = \sim ac$, and similarly for their inverses.

We will return to the algebraic analysis after an example (see Figure 1).

Let us take A to mean 'place A is full' and $\tilde{\text{A}}$ to mean 'place A is empty', and similarly for B. Then the moving of a block from B to A, presuming A is empty, would be

$$(\tilde{A}, B) \xrightarrow{a} (\tilde{A}, \tilde{B}) \xrightarrow{c} (\tilde{B}, \tilde{A})$$

$$\xrightarrow{a} (\tilde{B}, A) \xrightarrow{c} (A, \tilde{B}) = (\tilde{A}, B)acac.$$

If we rewrite the block movement in terms of Full and Empty, we have

$$(E, F) \xrightarrow{a} (E, E) \xrightarrow{c} (E, E) \xrightarrow{a} (E, F) \xrightarrow{c} (F, E).$$

The longer one looks at this, the stranger it seems, but what is going on is that, aside from the 'time' element, the two places - given precisely the indistinguishability of the sensory inputs - are being permutated, i.e., rotated. But the time element, introduced by $ac = b$, is keeping track in a different dimension, namely the complex ($\sqrt{-1}$, and recall that $b^2 = \tilde{b}^2 = -1$), which is also undergoing rotation.

Note that the transition between two opposites (think +1 and -1) can be viewed as a rotation of an axis containing both. Figure 2 combines three axes - places A and B, and the Hand (which is carrying the time dimension as viewed from the action $\tilde{\text{A}}\text{B} \longleftrightarrow \text{A}\tilde{\text{B}}$). We can combine the three axes at their zero-points because this is completely arbitrary; the axes are orthogonal because the coordinates they express are completely independent, e.g., the state of place A says nothing about the state of place B.

The movement of the block from B to A rotates each of the three axes through 180°. If we thereafter moved the block back from A to B, they would each rotate again through 180°. Such a complete 'circle' would thus record 360° for each of the three axes, but due to the indistinguishability of A and B, we would record 720° for the spatial rotation, but only 360° for the action itself. Speaking physically, this corresponds to spin-$\frac{1}{2}$.

Put somewhat differently, we can be tempted, since A and B have opposite states, to place them on the same "AB" axis. In this view, which ignores both the fact that each of A and B has a legitimate *local* view which is 2-valent, and as well the indistinguishability introduced by sensorial equivalence, the rotation which accomplishes the interchange of the block's position, and then interchanges again, then takes 360° and not 720°. Thus ignoring these distinctions renders spin-$\frac{1}{2}$ needlessly mysterious.

This example makes clear that treating sensors as being mutually orthogonal brings great clarity to a detailed understanding of how an action works. We exploit this orthogonality later when we map the sensors to a vector space.

The case of 3-actions is more complicated, due to the odd number of elements. They too form an exclusion group, $\mathcal{X}^3$, and we can quickly calculate that $\mathcal{X}^3$ contains $2^3 3! = 48$ elements. Let us write down the three sets of 3-actions where, respectively, the A, B or C component is held constant - what we call '2+1' actions:

	1	$\tilde{1}$	a	$\tilde{a}$
$\mathcal{C}_A^2$:	(123)	$(1\tilde{2}\tilde{3})$	$(1\tilde{3}2)$	$(1\tilde{2}3)$
$\mathcal{C}_B^2$:	(123)	$(\tilde{1}2\tilde{3})$	$(32\tilde{1})$	$(\tilde{1}23)$
$\mathcal{C}_C^2$:	(123)	$(\tilde{1}\tilde{2}3)$	$(1\tilde{2}3)$	$(\tilde{1}23)$

	b	$\tilde{b}$	c	$\tilde{c}$
$\mathcal{C}_A^2$:	$(13\tilde{2})$	$(12\tilde{3})$	(132)	$(1\tilde{3}\tilde{2})$
$\mathcal{C}_B^2$:	$(12\tilde{3})$	$(\tilde{3}21)$	(321)	$(\tilde{3}2\tilde{1})$
$\mathcal{C}_C^2$:	$(\tilde{2}13)$	$(2\tilde{1}3)$	(213)	$(\tilde{2}\tilde{1}3)$

We see that, comparing with the table for $\mathcal{X}^2$ earlier, each of the above individually is in fact an example thereof, modulo the constant element. Taking $\mathcal{C}_C^2$ as an example, we find that $ab = (1\tilde{2}3)(\tilde{2}13) = (2\tilde{1}3) = c$ and $ba = (\tilde{2}13)(1\tilde{2}3) = (\tilde{2}\tilde{1}3) = \tilde{c}$, and hence $ab = -ba$. Similarly, $ac = -ca$ and $bc = -cb$, and finally, $a^2 = c^2 = +1$ and $b^2 = \tilde{1}$. Hence $\mathcal{C}_C^2$ is indeed an exclusion group, as are $\mathcal{C}_A^2$ and $\mathcal{C}_B^2$. However, each has its own concept of "-1". In this connection, we note that $\tilde{1}_A\tilde{1}_B = \tilde{1}_B\tilde{1}_A = \tilde{1}_C, \tilde{1}_B\tilde{1}_C = \tilde{1}_C\tilde{1}_B = \tilde{1}_A$, and $\tilde{1}_C\tilde{1}_A = \tilde{1}_A\tilde{1}_C = \tilde{1}_B$; and (e.g.) $\tilde{a}_A b_B a_C = (1\tilde{2}3)(12\tilde{3})(\tilde{1}23) = (\tilde{1}\tilde{2}\tilde{3}) = -1$.

We can furthermore observe that $\mathcal{C}_A^2 \cap \mathcal{C}_B^2 \cap \mathcal{C}_C^2 = (123)$, and that $\mathcal{C}_A^2 \cap \mathcal{C}_B^2 = (12\tilde{3}), \mathcal{C}_A^2 \cap \mathcal{C}_C^2 = (1\tilde{2}3)$, and $\mathcal{C}_B^2 \cap \mathcal{C}_C^2 = (\tilde{1}23)$, which is to say that the three groups overlap - sharing the same identity element and pairwise an additional element. Notice also that (e.g.) $b_A c_B c_A = (13\tilde{2})(321)(132) = (\tilde{2}13) = b_C$, which is to say that we can get $\mathcal{C}_C^2$ from products of elements of $\mathcal{C}_A^2$ and $\mathcal{C}_B^2$, or in general, given two of the groups, we can generate the third. The remainder of the 48 elements can be generated via the c's and -1's.

We have therefore demonstrated that the complete set of actions on three (given) sensors, that is, 3-actions, can be viewed as the product of all the possible 2+1-actions on these same three sensors:

Theorem. The exclusion group $\mathcal{X}^3 = \mathcal{X}_p^2 \barwedge \mathcal{X}_q^2$, where $\barwedge$ denotes the overlapping product of the two groups uncovered above[7], and p, q are different and chosen from $\{A, B, C\}$.

A similar analysis of 4-actions reveals

Theorem. The exclusion group $\mathcal{X}^4 = \mathcal{X}^3 \barwedge \mathcal{X}^3$.

[7] We use the symbol $\barwedge$ because we are as yet unwilling to commit to (say) the direct or semi-direct product, or something else entirely.

We conjecture that $\mathcal{X}^n = \mathcal{X}^{n-1} \barwedge \mathcal{X}^{n-1}, n \geq 3$.

We turn now to the aforementioned correspondence of orthogonal sensors and vector spaces.

4.3 Sensors, Co-Occurrences, and Flux

We consider a set of n sensors $(s_1, s_2, \ldots, s_n)$, all fixed symbols, and define a *valuation* v on sensors by $v(s_i) = \sigma_\alpha$, where σ_α is one of $\sigma_1, \sigma_2, \ldots, \sigma_z$. Here take $z = 2$ and (with an eye to future use) take $\sigma_1 = 1, \sigma_2 = -1$. As noted earlier, with a single value $(z = 1)$ we would only get permutations, whereas with two values, we get rotation combined with change and hence *orientation*.

To maintain consistency with later analysis, we interpret $s = 0$ to mean a sensor value which cannot occur. The *orientation* of a sensor vector is positive when $\sigma_i > 0$ and negative when $\sigma_i < 0$.

We next impose a vector space structure S over the ring $\mathbb{R}$ of reals[8] on sensors, so $(s_1, s_2, \ldots, s_n)$ is a basis of S and any element of S is $s = \sum_{r=1}^{n} x_r s_r$. Here $+$ is purely formal.

In this connection, we have decided that the proper interpretation of "+" is "co-occurrence", in that

- The actuality of +'s components is indicated by their very presence, or rather, their absence can be ignored;
- Order is unimportant, or rather, the absence of order is necessary;
- Those sensor values which can co-occur correspond to those which can change or be changed in parallel;
- The additive identity "0" is interpreted to mean "cannot not occur", so reasoning involving equality, for example $X = Y$, which means $X + (-Y) = 0$, comes to mean that X and its inverse cannot co-occur, ie. they exclude each other;
- The connection of "+" to the addition of numbers is fundamentally irrelevant - the environment can choose to view things as having been "added" or not, as it likes. Another way to view this is that addition of simple magnitudes occurs in the algebra of the real numbers, which is beneath the level of the present reasoning.

We now extend valuation to S by $v(s) = \sum x_r v(s_r)$. Since we are now interpreting $+$ as co-occurrence, transformations leaving $+$ invariant (linear transformations) are important.

We assume an inner product "$\cdot$" between the elements of S, such that $s_i \cdot s_j = 1$ when $i = j$ and zero when $i \neq j$. The inner product expresses the projection of one vector on another, and hence the fact that the inner product of two different sensors is zero means, as noted earlier, that a given sensor in principle says nothing about any other. (It is of course co-exclusion's job to detect and abstract over the correlations between sensors, including those that might directly overlap.)

Now consider changes in the values of v for the various sensors: a "flux". The purpose of constructing S was to give a new *computational* way of talking about fluxes. If $v(s_i)$ changes from 1 to -1, let us write $s_i \to \tilde{s}_i$ where $v(\tilde{s}_i) = -v(s_i) = v(-s_i)$, so that when $v(s_i) = -1$, $v(\tilde{s}_i) = 1$. Call $\tilde{s}_i$ the complementary sensor. Make the convention that all sensors written down have $v = 1$ so that $s_1 + s_2 \to s_1 + \tilde{s}_2$ means that initially $v(s_i) = v(s_j) = 1$, but then $v(s_2)$ changes to -1. In this way a flux is "taken into the algebra". Note that $-s_i$ is interpreted as $\tilde{s}_i$. The interpretation given to ks_i, $k \neq \pm 1$ is 'stone age binary', eg. $2s = s + s$ is a co-occurrence of two distinct but otherwise indistinguishable s's.

4.4 Clifford Algebras and Vector Spaces

So far we have an algebra of co-occurrences. The various group properties of actions that we saw earlier prompt the choice of a *Clifford algebra* as the next step. As we will soon see, *products* in this algebra, not to mention other aspects, fit these properties extremely well.

It is useful to have a compact definition of what a Clifford algebra is. The following follows [ChoDe82].

Definition. Let $S^n_{(r)}, r \in Z^+, r \leq n$ be an n-dimensional vector space over the real numbers with inner product "$\cdot$" and basis (e_i) such that

$$\begin{array}{ll} e_i \cdot e_j = 0 & i \neq j \\ e_i \cdot e_j = -1 & i = j = 1, \ldots, r \\ e_i \cdot e_j = +1 & i = j = r+1, \ldots, n \end{array}$$

The characterization of S^n according to r follows from the fact that some of the basis vectors may have square -1; thus r is zero for our basic sensor level (thought not necessarily for the meta-levels). Using for example (a, c) as the basis, $a^2 = c^2 = 1$ but $b^2 = \tilde{1}$, and the group $\mathcal{X}^2$ is covered exactly by $S^2_{(0)}$. The notation $\{r, n-r\}$ expresses the *signature* of the space, eg. standard $3+1$-dimensional space-time would have $n = 4, r = 1$ and hence signature $\{1, 3\}$. In the following, we occasionally assume that $r = 0$ to avoid notational clutter; this means that some of the formulas will otherwise need a minus-sign.

If we define a product uv on the vectors in $S^n_{(r)}$ which is associative and distributive with respect to addition and which satifies the condition $uv + vu = 2(u \cdot v)$, i.e., $uv = -vu$ when u, v are orthogonal, then the resulting algebra of all possible sums and products is called the *Clifford algebra* $C(S^n_{(r)})$, which we will abbreviate as $\mathcal{C}^n_{(r)}$, and as $\mathcal{C}^n$ whenever the value of r is clear. It follows from these definitions that the algebra of $\mathcal{X}^2$ is $\mathcal{C}^2$.

Clifford algebras[9] can express a large number of fundamental and crucial aspects of physical reality,

[8] We may only need the rationals, but for now assume the worst.

[9] Grassman or 'exterior' algebras are similar, except that $|ee| = 0$ instead of 1, due to the fact that the product ee is solely 'outer', rather than 'inner + outer' as in a Clifford al-

which we will discuss later. For the present, we will simply remark that the phrase "all possible sums and products" of the vectors in $S^n_{(r)}$ corresponds to all possible state and action configurations in a co-exclusion based program. The intention of the following is to sketch how Clifford algebras can provide a good (and hopefully in the fullness of time, complete) formal characterization of phase web semantics.

A Clifford algebra is itself a linear space of dimension

$$\sum_{p=0}^{n} \binom{n}{p}$$

with basis $(1, e_{J_1}, e_{J_1}e_{J_2}, ..., e_1e_2\cdots e_n)$, where the J_j label ordered natural numbers, $J_j < J_{j+1}$. The semantics which we present below is that of a Clifford algebra over all possible sums and products of the combinations of the the states that S^n provides, i.e., all possible sums and products of

$$s_1,\ s_2,\ \ldots, s_n,\ s_1s_2,\ s_1s_3,\ \ldots,\ s_1s_n,$$

$$s_2s_3,\ \ldots,\ \ldots,\ s_1s_2\cdots s_n$$

Hestenes [HeSo] generalizes Clifford algebras into a general calculus of space-time, called the *geometric calculus*.

4.5 Expressing Action

The purpose of constructing a Clifford algebra is two-fold. Firstly, the linear functions on S (i.e. transformations preserving co-occurrence) are of the form

$$s \to f(s) = \sum_{\alpha=1}^{2^n} a_\alpha s b_\alpha$$

where a_α, b_α are elements of the algebra i.e. expressions of the form

$$a_\alpha = a_{\alpha,0} + \sum_i a_{\alpha,i}s_i + \sum_{i<j} a_{\alpha,ij}s_is_j + \sum_{i<j<k} a_{\alpha,ijk}s_is_js_k + \ldots$$

Secondly, as a special case, either the a_α or the b_α can be taken to be the 2^n units

$$1, s_i, s_is_j, \ldots, s_1s_2...s_n$$

themselves[10]. The latter forms allow us to entertain the idea that products can represent *actions*. See also Figure 3.

The vertical dimension in Figure 3 corresponds to the product $s_1s_2 = \frac{1}{2}(s_1 + s_2)(s_1 + \tilde{s}_2)$. The 2-dimensional vector space S^2 is in fact uniquely determined by the non-zero 2-blade s_1s_2 [HeSo, p.17]. There is an implicit assumption throughout this presentation that the underlying space is Euclidean, but this can presumably be avoided.

Summing up the development to this point, we have an algebra which can express

- The existence of sensors and their changing values as a vector space;
- The co-occurrence of sensors as one of its fundamental operations: + ;
- The concept of mutual exclusion as its zero;
- The anti-commutativity of actions using its fundamental product $s_1s_2 = -s_2s_1$;
- The effect of these actions on co-occurrences preserves +, for example, it makes no difference whether we think of an action as $(s_1 + s_2)s_3s_4$ or as $s_1s_3s_4 + s_2s_3s_4$.

So far so good. We lack however a means of expressing exactly how the algebra's product expresses the phase web's concept of action. What we are looking for is a linear transformation which preserves products, i.e. an automorphism of $\mathcal{C}(S)$. If S contains an even number of elements, then the required automorphism is *inner*, that is, of the form

$$s \to s' = asa^{-1}$$

[In general, Clifford algebras exhibit various quirks when the basis contains an odd number of elements. Regarding the above automorphism, for *odd* algebras an automorphism is either inner, or it is the composition of an inner automorphism with $i \to -i$ (complex conjugation).]

Consider as an example action the case $n = 2$, and look at $s_1 + s_2$. The possible changes are:

$$s_1 + s_2 \to s_1 + s_2$$

$$s_1 + s_2 \to s_1 + \tilde{s}_2$$

$$s_1 + s_2 \to \tilde{s}_1 + s_2$$

$$s_1 + s_2 \to \tilde{s}_1 + \tilde{s}_2$$

Since $s_1^{-1} = s_1$, and we find

$$s_1(s_1 + s_2)s_1 = s_1 + \tilde{s}_2$$

and also $s_2^{-1} = s_2$, so

$$s_2(s_1 + s_2)s_2 = \tilde{s}_1 + s_2$$

and finally, since $(s_1s_2)^{-1} = s_2s_1$, a genuine *action* has the form

$$s_1s_2(s_1 + s_2)s_2s_1 = \tilde{s}_1 + \tilde{s}_2$$

Similarly, for 3- and 4-ary actions, we get

$$-s_1s_2s_3(s_1 + s_2 + s_3)s_3s_2s_1 = \tilde{s}_1 + \tilde{s}_2 + \tilde{s}_3$$

gebra. Clifford algebras are a generalization of Grassman algebras, Hamilton's quaternion algebra, tensor algebra, and vector manifolds.

[10]This theorem is due to Kilmister [1950], but the result for quaternions was known in the 1930's by A.W. Conway (no relation to J. Conway).

where the minus sign reflects the effect of the complex conjugation mentioned above, and

$$s_1 s_2 s_3 s_4 (s_1 + s_2 + s_3 + s_4) s_4 s_3 s_2 s_1 = \tilde{s}_1 + \tilde{s}_2 + \tilde{s}_3 + \tilde{s}_4$$

Notice that the structure of the inner morphism contains the 'action' twice, reflecting what we noticed in the example of moving the block (cf. Figure 1, where the operation *ac* must be done twice). Furthermore, we have seen that a 2-action is exactly captured by the product $s_i s_j$ (taken twice), and also (earlier) that 3-actions can be expressed as products of 2+1-actions, etc. So now we know how to express actions in our algebra.

Finally, this apparatus leads to two composition rules for actions, where by 'composition' we mean that one action acting on the result of another can be viewed as a single combined action.

Suppose we have the actions $s_1 + s_2 \rightsquigarrow \tilde{s}_1 + \tilde{s}_2$ and $\tilde{s}_2 + s_3 \rightsquigarrow s_2 + \tilde{s}_3$, where it is the $\tilde{s}_2$ in the second action that expresses that it builds on the result of the first, and which therefore allows it to be composed with it. From above we know

$$s_1 s_3 (s_1 + s_3) s_3 s_1 = \tilde{s}_1 + \tilde{s}_3$$

We can rewrite the righthand side as

$$s_1 s_3 (s_1 + s_3) s_3 s_1 = \tilde{s}_1 + (\tilde{s}_2 + s_2) + \tilde{s}_3$$

since $\tilde{s}_i + s_i = 0$. In terms of our interpretation, this substitution means that, since the two valuations of s_2 cannot co-occur, they must occur in some order or other. We choose the order

$$s_1 s_3 (s_1 + s_3) s_3 s_1 = (\tilde{s}_1 + \tilde{s}_2) + (s_2 + \tilde{s}_3)$$

in that we substitute with the two terms' inner automorphisms, yielding

$$s_1 s_3 (s_1 + s_3) s_3 s_1 = s_1 s_2 (s_1 + s_2) s_2 s_1 + s_2 s_3 (\tilde{s}_2 + s_3) s_3 s_2$$

which is the desired composition rule[11].

In a similar fashion, it is easy to derive

$$-s_1 s_2 s_3 (s_1 + s_2 + s_3) s_3 s_2 s_1 - s_3 s_4 s_5 (\tilde{s}_3 + s_4 + s_5) s_5 s_4 s_3$$

$$= s_1 s_2 s_4 s_5 (s_1 + s_2 + s_4 + s_5) s_5 s_4 s_2 s_1$$

where the two actions $s_1 s_2 s_3$ and $s_3 s_4 s_5$ overlap on s_3, that is, a single sensor; and

$$-s_1 s_2 s_3 (s_1 + s_2 + s_3) s_3 s_2 s_1 - s_2 s_3 s_4 (\tilde{s}_2 + \tilde{s}_3 + s_4) s_4 s_3 s_2$$

$$= s_1 s_4 (s_1 + s_4) s_4 s_1$$

where they overlap on two sensors. The pattern should be clear by now - the overlapping, complementary sensors simply drop out, leaving the rest.

The second composition rule was suggested by C.W. Kilmister [Kil92]. Given

$$s_1 s_2 (s_1 + s_2 + s_3) s_2 s_1 = \tilde{s}_1 + \tilde{s}_2 + s_3$$

and

$$s_2 s_3 (s_1 + s_2 + s_3) s_3 s_2 = s_1 + \tilde{s}_2 + \tilde{s}_3$$

implies

$$[s_2 s_3] s_1 s_2 (s_1 + s_2 + s_3) s_2 s_1 [s_3 s_2] = \tilde{s}_1 + s_2 + \tilde{s}_3$$

which is the result of $s_1 s_3$.

4.6 Co-Exclusion and the Boundary Operator

This section exhibits the explicit connection between the basic co-occurrence/co-exclusion inference of the phase web paradigm and the mathematical apparatus we have presented thus far.

To begin, we can express the co-exclusion principle formally for 2-actions as:

The ***co-exclusion principle*** says that if we observe $(s_1 + s_2)$ *and* $(\tilde{s}_1 + \tilde{s}_2)$, then it is correct to conclude that $(s_1 + s_2)xyxy = (\tilde{s}_1 + \tilde{s}_2)$, where $x : (s_i, s_j) \rightarrow (s_j, s_i)$ and $y : (s_i, s_j) \rightarrow (s_i, \tilde{s}_j)$.

Example. In terms of our earlier analysis, $xyxy$ corresponds to $caca = bb$. Suppose therefore we observe $(s_1 + s_2)$ and its inverse. Then

$$(s_1, s_2)xyxy = (s_1, s_2)(2\tilde{1})(2\tilde{1}) = -(s_1, s_2) = (\tilde{s}_1, \tilde{s}_2)$$

We find therefore that $(s_1 + s_2)xyxy = (\tilde{s}_1 + \tilde{s}_2)$ as expected.

We can conclude from this formulation of co-exclusion that if we take the primitive sensor values as real numbers, the operator $xyxy = (xy)^2$ will have the value -1. The significance of this conclusion is that we can *reconstruct the world*, which experience has shown requires $i = \sqrt{-1}$, on the basis of real information.

A persistent intuition has been that, given that the paradigm is so intimately connected with *change*, the concept of differentiation must have a role to play. Our first foray in this direction, inspired also by the geometric flavor of various things we have presented, is to look at algebraic topology, more specifically, homology theory.

It turns out that we can get the equivalent of (one concept of) differentiation via homology theory's boundary operator, ∂. Define

$$\partial(s_{i_1} s_{i_2} \ldots s_{i_k}) = \sum_{r=1}^{k} (-1)^{r+1} s_{i_1} \ldots s_{i_{r-1}} . s_{i_{r+1}} \ldots s_{i_k}$$

for $k > 1$, and for $k = 1, \partial(s_i) = 1$ (rather than 0 - arbitrarily chosen - as in the usual theory). Then

$$\partial(a + b) = \partial a + \partial b$$

[11] The insertion of $s_2 + \tilde{s}_2$ can also be viewed as an expression of the fact that 'anything can happen' while an action is being carried out, inasmuch as the action does not in itself say anything about such matters. The composition rule thus states that one may insert any number of such intermediate steps. Nevertheless, one wonders if there might not be something special after 137 steps, cf. [McGNo, p.97, middle paragraph, right column]

$$\partial(\partial a) = 0$$

and

$$\partial(ab) = (\partial a)b + (-1)^o a(\partial b)$$

where o is the order of a (i.e. if a is a p-simplex, $o = p+1$), just like the differentiation of exterior products.

As a simple geometric example of the boundary operation, consider an ordinary triangle ABC, where we specify it in terms of its vertices A, B, C and its edges are thus AB, BC, CA. Then

$$\partial(ABC) = BC - AC + AB$$

Since specifying the triangle's edges in terms of the vertices means that edge AC is oriented oppositely to edge CA, we can rewrite the above as $AB+BC+CA$, which is indeed the boundary of the triangle.

Notice, by the way, that the fact that a triangle's components are *discrete* entities plays no role. Indeed, despite the ingrained association of 'continuity' with 'differentiation', there is a surprising *isomorphism* theorem between the operations of the exterior differential calculus and the homology theory's boundary operator (cf. DeRham's theorem). This also puts [PeSm], which proves continuity over partial orderings (read, discrete concurrent events), into context.

We now ask the question, "What is the boundary of $s_1 s_2$", or rather, for reasons which will become quickly apparent, "What is $\partial(s_1 s_2 = -s_2 s_1)$?" We know that $s_1 s_2 + s_2 s_1 = 0$, and working out ∂ of this we find

$$s_2 - s_1 + s_1 - s_2 = 0$$

Re-writing this in terms of our valuation conventions, this becomes

$$s_2 + \tilde{s}_1 + s_1 + \tilde{s}_2 = 0$$

and, given that + is commutative, we can rearrange this in two different but equivalent ways:

$$(s_1 + \tilde{s}_2) + (\tilde{s}_1 + s_2) = 0$$

and

$$(s_1 + s_2) + (\tilde{s}_1 + \tilde{s}_2) = 0$$

which is to say that $(s_1 + \tilde{s}_2)$ may not co-occur with $(\tilde{s}_1 + s_2)$, nor may $(s_1 + s_2)$ co-occur with $(\tilde{s}_1 + \tilde{s}_2)$. Notice that these are the initial premises (both forms!) of the co-exclusion principle.

Now it is obvious that if $\partial b = 0$ then there is an element a such that $b = \partial a$. For, if not, this would give b as a 'cycle', corresponding to a 'hole' in the space (non-trivial homology) whereas the Clifford algebra has trivial homology. So, running the derivation in the opposite direction[12], we can conclude that ***the fact that a co-occurrence and its complement cannot co-occur means that the product of the component vectors is a Clifford product. In other words, the co-exclusion principle is really a statement about 'integrating' change to derive a whole, and the 'action' performed by such a whole is to change the orientation of its boundary!*** All this generalizes to co-exclusion inferences of arbitrary arity, and thus gives the original intuition a firm formal basis.

We can interpret this result further. Think back to the fact that the computational foundation of co-exclusion is mutual exclusion, and that when two processes mutually exclude each other there exists a resource invariant. The resource invariant expresses the fact that the synchronization 'stick' must move on a closed orbit, which orbit corresponds to the boundary expressed by the operator ∂. The fact that the resource invariant is indeed *invariant* is captured by the fact that $\partial(\partial(a)) = 0$, or as often put, "The boundary of the boundary is zero".

The remainder of this paper interprets the results above in terms of physics, and the implications for the construction of nano-scale hardware.

5 Where's the Physics?

The preceding mathematical formulation hopefully demonstrates, in spite of its spottiness and relative lack of rigor, that there is a firm connection between basic computational concepts - synchronization, co-occurrence, mutual exclusion, co-exclusion - and more or less ordinary physical mathematics. Moreover, this mathematics is in essence precisely that of modern physics - quantum mechanics and relativity theory. In the following, therefore, we appeal to these connections to make plausible what otherwise are relatively loose identifications between the above computational phenomena in the model and well-known physical phenomena.

Spin, fermions, and bosons. We identify the fact that $(s_1 s_2)^2 = -1$ with the quantum mechanical attribute called 'spin one-half'. The way this anti-commutativity is connected with the orientation of $s_1 s_2$, together with the fact that the transition $s_1 + s_2 \rightarrow \tilde{s}_1 + \tilde{s}_2$ both changes the orientation of $s_1 s_2$ and, being an action, changes the state of the system, yields the PT part of CPT invariance automatically. We identify therefore co-exclusion-based actions generally as fermions, even though (eg.) 4- and 5-actions have square +1. It should also be apparent how 'artificial' quantum numbers such as isospin are modelled.

Similarly, we identify the fact that $\frac{1}{n}(s_1 + s_2 + \ldots s_n)^2 = 1$ with spin one, that is, co-occurrences are bosons. We conjecture that the simplest of these, 2-co-occurrences, models photons and vector-bosons, and that 3-co-occurrences model gluons. We speculate that a 'graviton' is a meta-co-occurrence over $\geq$ 4-actions.

Particles & anti-particles. [Man92] describes how the computational concept of a resource invariant can be interpreted as a criterion for 'objecthood'. In particular, it implicitly advances the idea that a

[12] The inverse ∂^{-1} is not unique, which means that additional inferences are possible: $\pm k_a s_1 s_2 + k_b(\sum \text{other } s_i\text{'s, such that } \partial_\Sigma = 0) + k_c$. The $k_a \neq \pm 1$ coefficient is uninteresting, and the k_b and k_c terms merely state the possibility that a larger context is necessary to actually complete the definition of the action (eg. 2+1 actions). As it is, we know the basic inverse, and this is enough.

fermion should be viewed as a mutual exclusion structure, such as is represented by co-exclusion-based actions. It would seem to follow that the particle - anti-particle relation is that of $s_1 s_2$ to $s_2 s_1$, in that both the orientation and direction of change are reversed. In general, the present model implies that what are called 'particle' and 'anti-particle' are two states of the 'same' object that are 180^o out of phase. It follows that a primitive 2-action, being the simplest action, should be identified with a neutrino. Modelling the neutrino - anti-neutrino interaction (perhaps naively) as the co-occurrence $s_1 s_2 (s_1 + s_2) s_2 s_1 + s_2 s_1 (\tilde{s}_1 + \tilde{s}_2) s_1 s_2$ yields $(s_1 + s_2) + (\tilde{s}_1 + \tilde{s}_2) = \gamma^{LL} + \gamma^{RR}$, which can be interpreted as the two neutral vector bosons. See Figure 4.

We speculate that electrons are meta-2-actions, in that their additional properties require corresponding additional information-carrying capacity. As just indicated, a general property of the model is that 'generations' of particle 'families' arise via identical arities, but composed from different co- and/or meta-co-occurrences. Greater mass/energy is presumably indicative of greater information-carrying capacity.

Quarks. We identify 3-actions with quarks and mesons. The ubiquitousness three-ness of the quark particle family is the initial motivation, along with the the fact that 3-actions represent the next step 'up' in complexity. It is a natural consequence to identify the elements of the three 2+1-action groups of $\mathcal{X}^3$ with the various quarks and anti-quarks, the three families thereof. Along the same lines, the fact that quark-quark interactions flip two of the three 'colors' corresponds to the fact that each of the three $\mathcal{X}^2$ groups holds one of its three values constant when it changes state. It is our understanding that the various cardinals of these groups also fit this identification.

Finally, the holding of one value constant has a thought-provoking implication, best seen through an example. Imagine a very simple blocks world situation with one hand (H), two places (A,B) and one block, with sensors for hand position (HA, HB), hand content (CH), and place content (CA, CB). Assume the existence of the following co-exclusion based actions:

GA: (CA=1, CH=$\tilde{1}$)$\leftrightarrow$(CA=$\tilde{1}$, CH=1) ie. block move 'tween A and H; 1/$\tilde{1}$=full/empty.

RB: (CB=1, CH=$\tilde{1}$)$\leftrightarrow$(CB=$\tilde{1}$, CH=1) ie. block move 'tween B and H.

HAB: (HA=1, HB=$\tilde{1}$)$\leftrightarrow$(HA=$\tilde{1}$, HB=1) ie. hand move 'tween A and B; 1/$\tilde{1}$=@A/@B.

Given that the block is at A, ie. CA=1, CB=$\tilde{1}$, CH=$\tilde{1}$, the problem is to move the block to B. One imagines immediately that the proper sequence of actions is to grasp at A, ie. do GA, move the hand from A to B, do HAB, and release at B, do RB.

In order for an action to actually 'fire', its pre-conditions must be fulfilled, whereafter it will attempt to fulfill its post-conditions by issuing appropriate goals (cf. end of §3). In the case of GA, CA=1 and CH=1 so GA will succeed. But notice that GA will also 'fire' if the hand is at B, ie. HA=$\tilde{1}$, HB=1! That is, the 2-actions GA, RB given above lack a crucial piece of information: the hand must be at the appropriate place if the grasp or release action is in fact to succeed. Furthermore, this information about *commonality of place during the action* cannot be expressed by a 2-action, or even a 3-action in which all three components change. What is needed is a '2+1'-action, where two components change, and one (here, the position of the hand) does not: (CA=1, CH=$\tilde{1}$, *HA=1*) $\longleftrightarrow$ (CA=$\tilde{1}$, CH=1, *HA=1*).

That certain kinds of change *require* the expression of some dimensional commonality between the 'before' and 'after' states prompts us to see herein further growth of co-occurrence's seed of the concept of 'space', although this seed requires the emergence of yet further hierarchical structure to reach full expression in the form of ordinary 3-dimensional space. At the same time, the corresponding constancy of one of the three components of the 2+1-action is exactly what we find in the members of the $\mathcal{X}^3$ group we have identified with quarks.

Quaternions. Quaternions are a group of three operators that express rotation in three dimensions. Their defining properties are: $ab = -ba, bc = -cb, ca = -ac; ab = c, bc = a, ca = b; (ab)^2 = (bc)^2 = (ca)^2 = -1$. They appear as the boundaries of the 3-action abc, ie. $\partial(abc) = bc - ac + bc$. Hence the meta-3-action created by co-excluding $a = s_1 s_2, b = s_2 s_3, c = s_3 s_1$ implicitly contains, as it were, the ability for the corresponding 3-object to 'understand' 3-dimensional space from its own point of view. Full-blown 3+1 dimensional space-time will presumably require four tetrahedrally-related such 3-objects to appear.

It also bears mentioning that the quaternion algebra is an even sub-algebra of the Pauli algebra, which in turn is an even sub-algebra of the Dirac algebra, and all are Clifford algebras; see [ChoDe82] for details. In the same breath, however, we point out that the relationship established by ∂ is not that of an algebra to an even (or odd) sub-algebra.

Causality. Both of the action-composition rules given in the preceding section can be viewed as providing models of causality, ie. the necessary sequence in which actions can be performed. We prefer the first of them, in that the 'de-composition' $s_1 s_3 \mapsto s_1 s_2 + \tilde{s}_2 s_3$ models the cascade of requests (mentioned at the end of §3) and allows for the instigation of concurrent and non-deterministic activity. The second of the composition rules is in contrast very sequential and deterministic in its thrust. Howsoever, the arity of the actions involved, which can be viewed as a hierarchical issue, determines whether the causal influence is classical momentum transfer or EPR's more ethereal version.

Formal Structure. Although we have only hinted at it, it appears quite certain that the basic overall mathematical structure of the theory we have presented is that described by homology theory. The

'twisted isomorphism' between homology and cohomology seems particularly worth further investigation.

Closer to what we have presented is the matter of the *commutator* product: $A \times B = [A,B] = \frac{1}{2}(AB - BA)$, where A, B are arbitrary multi-vectors. This product satisfies the following relationships (see [HeSo, p.14] for the first four):

bi-linearity: $[\lambda A + \mu B, C] = \lambda[A,C] + \mu[B,C]$,

$$[A, \lambda B + \mu C] = \lambda[A,B] + \mu[A,C];$$

anti-commutativity: $[A,B] = -[B,A]$

Jacobi ident: $[A,[B,C]] + [B,[C,A]] + [C,[A,B]] = 0$;

Leibniz I: $[A,BC] = [A,B]C + B[A,C]$;

Leibniz II: $\partial[A,B] = [\partial A, B] + [A, \partial B] + \partial(AB)$.

One of the hallmarks of quantum-mechanical behavior is $[A,B] \neq 0$, which is true of both 2- and 3-actions. More interestingly, Dyson [Dy90] presents a derivation, originally due to Feynman (1948), of Maxwell's laws from Newton's laws plus the commutation relations between position and velocity of a single non-relativistic particle. Tanimura [Tan92] generalizes this derivation to encompass both special and general relativistic contexts. According to Tanimura, besides Newton's laws as appropriate (and there is no need of the a priori existence of a Hamiltonian, Lagrangian, canonical equation, or Heisenberg's equation), all the derivations appeal only to the *form* of the commutator product, and are completely independent of the interpretation placed on its components. In this connection, it is interesting that Kaufmann [Kauf87] shows that the one needs only parenthesis-operations and component-ordering to arrive at both spinors and the Lorentz group.

It thus appears that the only things separating our framework from a ∂-based expression of Maxwell's laws are (1) we must show that Newton's laws apply, and (2) Tanimura's form of Leibniz II requires that our $\partial(AB) = 0$. We expect that (2) will fall into place naturally, whereas the fate of (1) depends on properties which we expect will first emerge with the appearance of ($\geq$)4-meta-actions, namely 'ordinary' space-time.

Space and Time. The availability of these concepts rests on one of the more radical implications of the model being presented here, namely the fact that the usually primitive concepts of 'ordinary time' and 3-dimensional space are neither given nor assumed a priori, but rather *emerge* as the hierarchical structure expands. The reason why this is so, simply stated, is that to give these concepts semantic content requires a context with sufficient information-carrying capacity to express the required distinctions. Such capacity accrues solely via (cycle-)hierarchical aggregation. The same reasoning applies to charge and mass: they are emergent attributes and hence will appear in due course as the information-carrying capacity of the constantly growing structure evolves. Howsoever, one can loosely identify 'co-occurring' with 'space-like separated', and 'co-excluding' with 'time-like separated'. The present approach appears to be isomorphic to that of the Combinatorial Hierarchy [BK, McGNo]. The Combinatorial Hierarchy is, in our view, of fundamental conceptual significance.

Summarizing this section, we believe we have made a reasonably detailed and plausible case for the position that the computational structure sketched in §I.2-3 captures the essential aspects of quantum mechanical systems. Indeed, we believe it represents the (neo-)mechanistic model of QM that has hitherto been lacking, and therewith that this structure in execution will be able to provide a blow-by-blow account of Feynman's sum-over-paths formulation. It also seems reasonable to believe that relativistic considerations are accomodated.

6 Making Quantum Machines

How then might these insights be transformed into a nano-scale computer architecture?

The initial point of departure is to recognize that at the nano-scale, quantum mechanical effects in the devices will be prominent, and therefore it will be foolhardy to fight against them. Rather, they must be exploited, made to work *for* us. So the question becomes, how can quantum-mechanical effects be harnessed to provide computation? The execution model sketched very briefly at the end of §3 has the following properties:

- Since *everything* is a process, there are *many* processes and no 'data structures';
- These processes exhibit global coherence with *no* centralized locus of control;
- They synchronize (rather than communicate) via a distributed global 'memory' a la Linda's *tuple space* [Gel85];
- They execute utterly concurrently and opportunistically;
- The execution regime is reversible, non-deterministic and goal-oriented;
- Duplicates of a given action or synchronization tuple are no problem;
- Systems with disjoint or compatibly defined bases ('sensor sets') can be combined ('composed') with (discardable) emergent effects;
- Conversely, a given system can be 'encapsulated' to provide modular 'black box' functionality to other systems;
- An initially 'blank' system can be 'trained' to provide a given behavior;

In general, the model provides a different style of computation from what we are used to: 'behavior' rather than 'transformation', although we expect that more traditional forms of computation can be accomodated.

Our implementation of the phase web model - in our local C++Linda - has provided proof-of-principle but is otherwise much too slow and cumbersome to be of any use (and we are therefore dis-inclined to 'release' it). Nevertheless, we have learned much from it, among other things that it takes something like Linda to implement it, that is, a linguistic vehicle which does not implicitly force its user to adopt a pre-ordained concept of 'parallel' computation. The particular property of Linda's model that is particularly helpful is its associative style of process interaction and the space-time decoupling this engenders. Presumably, any model providing this property would be suitable. In any event, we needed the following primitives:

- Tuple space - that is, a global synchronization medium;
- out(*tuple*) - emit a tuple to tuple space;
- rd(*tuple*) - detect the presence of a given tuple in tuple space;
- in(*tuple*) - remove a given tuple from tuple space under mutual exclusion;
- eval(*process*) - create a new process; this might be avoidable for systems that are not to be 'trained';
- CO($tuple_1$, $tuple_2$, ...) - detect a co-occurrence of the given tuples in tuple space, block until this is true;
- NotCO($tuple_1$, $tuple_2$, ...) - block while the given tuples co-occur in tuple space.

The trick, then, is to realize these primitives with quantum mechanical phenomena. The following is speculative. It would seem that quantum non-locality is a possible means of achieving tuple space's coordination, since we require only synchronization (which is non-information bearing) and not communication (which is). Tuples are naturally bosons, as of course are co-occurrences. The production, detection, and removal of tuples to/in/from tuple space correspond accordingly to the emission, detection, and absorption of bosons. Processes are actions are fermionic phenomena, eg. spin flips. Blocking behavior translates similarly into maintaining a given quantum state until disturbed - presumably by the energy available carried by a boson (cf. the Coin Demonstration). The overall idea is that computation is modelled by the changing phase relationships between the components of the system; this is the reason we refer to our model as the phase web paradigm.

7 Conclusion

We have argued that computation - when it includes synchronization explicitly - *can* model quantum mechanical and relativistic phenomena at a fundamental level. The basic tools of that argument - synchronization and co-occurrence - also lead to a model of execution that is described naturally by the same mathematical structures as describe the relevant physics. This same model of execution appears to be realizable using the physical phenomena that will dominate nano-level hardware. The goal of nano-scale hardware apparently requires truly distributed computation, at the price of having to jettison our sequentially-infected prejudices of how a computer 'ought' to be. But since Nature *is* truly distributed, this is probably the right price to pay.

The reasoning and arguments leading to this conclusion are subtle, as are the issues, and given their importance it is unfortunate that they had to be presented in such compressed form. We do not expect everyone will be convinced immediately, and indeed to conduct the argument fully rigorously will presumably take the remainder of the decade. Skeptics and the interested reader are therefore also referred to [Rosen] for a parallel category-theoretical argument in an entirely different context - living systems - which we believe is actually focusing on the same fundamental issues.

I would greatly appreciate, if/when someone writes a paper referencing this one or the ideas contained herein, their sending me a (paper) copy. Software implementing the Phase Web paradigm is currently not available (August, 1994); inquire via http://www.iesd.auc.dk/general/DS/index.html or ftp.iesd.auc.dk/pub/reports/papers/manthey:X where X = PhysComp94.ps.Z for this paper, and PhaseWeb_Software.README for software status.

Acknowledgements. I would like to thank my students for their energy and enthusiasm; and my colleagues and friends - Clive Kilmister, H. Pierre Noyes, Keith Bowden, Eddie Oshins, Lou Kauffman, and others - in the Alternative Natural Philosophy Association (ANPA) for their help and support. This paper is dedicated, with the greatest respect, to Carl Adam Petri.

References.

Bastin, T. and Kilmister, C.W. *Combinatorial Physics.* (To appear).

Choquet-Bruhat,Y, DeWitt-Morette,C, et al. Analysis, Manifolds and Physics (revised edition). North-Holland, 1982. (Vol.1, pp.64-69)

Dyson, F.J. Feynman's Proof of the Maxwell Equations. American Journal of Physics (58)3, 1990. Pp. 209-210.

Feynman, R. *The Character of Physical Law.* Published by the British Broadcasting Corporation (BBC), 35 Marleybone High Street, London. 1965.

Gelernter, D.H. Generative Communication in Linda.

ACM Toplas (7)1. 1985.

Hestenes, D. and Sobczyk, G. *Clifford Algebra to Geometric Calculus.* Reidel Publishing Co. 1985.

Hestenes, D. *Foundations for Classical Mechanics.* Reidel Publishing Co. 1989.

Hewitt, C. The Challenge of Open Systems. Byte magazine, September, 1985.

Kaufmann, L. Special Relativity and a Calculus of Distinctions. Proceedings of ANPA West, 1987.

Kilmister, C.W. Personal communication.

Manthey, M. Synchronization: The Mechanism of Conservations Laws. Physics Essays (5)2, 1992.

McGoveran, D. and Noyes, H.P. An Essay on Discrete Foundations for Physics. Physics Essays (2)1. 1989.

Noyes, H.P. A Novel Theory of Everything. (This proceedings.)

Parker-Rhodes, F. *The Theory of Indistinguishables.* Reidel, 1981. ISBN 90-277-1214-X.

Penrose, R. *The Emperor's New Mind.* Oxford Press. 1989.

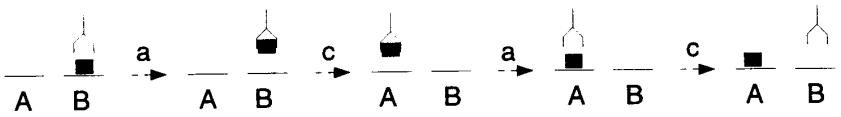
Petri, C.A. and Smith, E. Concurrency and Continuity. In *Advances in Petri Nets*, Springer LNCS 266, G. Rozenberg, Ed. 1987.

Phipps, T.E. Jr. Proper Time Synchronization. Foundations of Physics, v21,9. September, 1991.

Rosen, R. *Life Itself - A Comprehensive Inquiry into the Nature, Origin, and Fabrication of Life.* Columbia University Press, 1991. ISBN 0-231-07564-2.

Tanimura, S. Relativistic Generalization and Extension to the Non-Abelian Guage Theory of Feynman's Proof of the Maxwell Equations. Annals of Physics 220 (229-247). 1992.

(a) View from the table: the hand moves.



(b) View from the hand: the table moves.

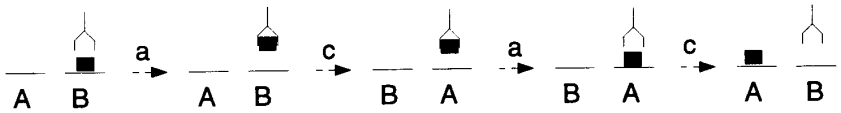


Figure 1: Block Movement from Two Points of View.

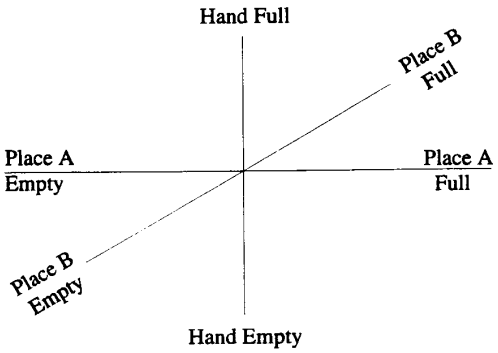


Figure 2: Moving a Block Requires Three Axes.

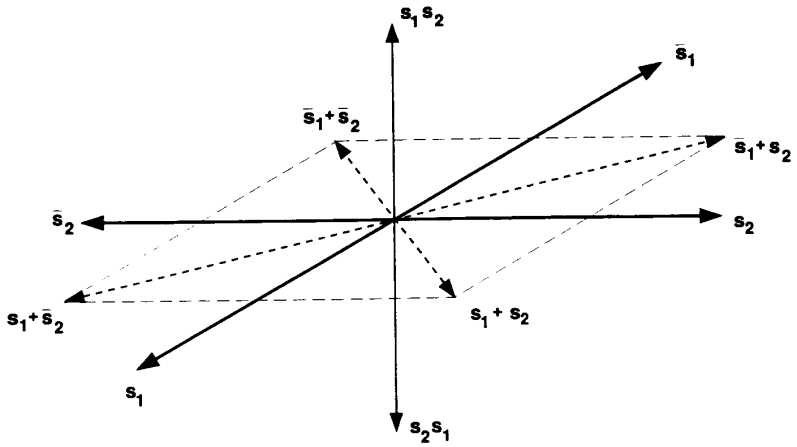


Figure 3: Two Possible Actions: $(s_1 + s_2) \leftrightarrow (\tilde{s}_1 + \tilde{s}_2)$ and $(\tilde{s}_1 + s_2) \leftrightarrow (s_1 + \tilde{s}_2)$.

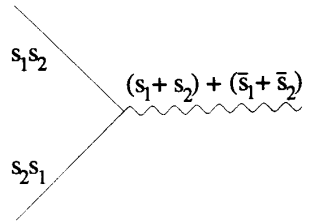


Figure 4: Feynman diagram of a $\nu/\tilde{\nu}$ interaction yielding two vector bosons.