ON THE SPECIFICATION OF CONCURRENT SYSTEMS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Ross Thomas Casley
January 1991

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Vaughan Pratt
(Principal Advisor)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

José Meseguer
SRI International

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

John C. Mitchell

Approved for the University Committee on Graduate Studies:

Dean of Graduate Studies

Preface

In order to make the presentation more self-contained, this dissertation contains some material that is well-developed in the mathematical literature but unfamiliar to computer scientists. Sections 2.1, 2.2, 2.4 and 3.1 are such introductory material, as is the definition of $F!$ in 4.1.

This dissertation also contains some results that have been previously published in jointly-authored work [13]. Except as otherwise noted in the text the results presented here are my own work, under the guidance of my advisor, Professor Vaughan Pratt.

Acknowledgements

I would like to thank my advisor, Vaughan Pratt. His ideas, advice and critical reading have contributed a great deal to this work. Roger Crew and José Meseguer, as collaborators, have influenced it in many ways. John Mitchell provided useful comments on rather short notice. Conversations with Bill Lawvere, John Power, Bob Walters and Mike Johnson have been very helpful.

I am very grateful to my wife, Mei Lin, for her support, encouragement and patience throughout the years I have been working at this dissertation.

This research was supported in part by the National Science Foundation under grant CCR–8814921.

Contents

Abstract

In models of concurrent processes constraints on the order of events are often represented by partial orders, and schedules of events are then defined using an algebra of standard operations such as sequential and parallel composition.

In this dissertation the notion of partial order is replaced by that of a set with a metric which takes values in a given ordered monoid. Partial orders are the simple case of a monoid whose two elements represent the presence or absence of a constraint.

An ordered monoid can be seen as a monoidal category, and schedules based on it are categories enriched in the monoid. Algebraic operations on schedules can then be defined as constructions in the category of schedules.

These definitions rely on certain properties of a category of schedules, such as closure and completeness. To simplify proofs of these properties, two constructions are defined. The first creates a category of unlabeled schedules from a system of constraints. The second adds labels to unlabeled schedules. Many categories of interest can be constructed from simple categories using these two methods. The main results of the dissertation derive the required properties of categories so constructed from similar, more easily verified properties of the base categories.

Several notions of timing constraint can be viewed in a uniform way in this framework. An example is the Gaifman-Pratt system, essentially the partial order model with additional specification as to whether two events may occur simultaneously. It corresponds to a monoid whose three elements represent strict precedence, lax precedence (simultaneity is permitted), and absence of constraint. Real-valued timing constraints correspond to the additive monoid of the real numbers.

Chapter 1

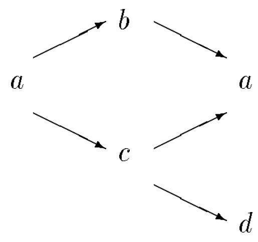
Algebras of Labeled Measured Sets

In this chapter measured sets and operations for combining them are described informally, using the partial order model of concurrency as motivation.

1.1 Outline

In the pomset model of concurrent processes [24, 22, 42], a process is specified by a set of *schedules* of events. Schedules are formally defined as *pomsets* (partially ordered multisets), consisting of a poset with labels on each element. Each label specifies a particular action. The partial order relation constrains the order in which events occur.

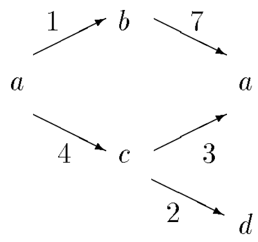
As an example, consider the pomset represented by the following Hasse diagram, in which arrows run from smaller to larger elements in the partial order and each element is represented by an occurrence of its label. It denotes a poset of five elements each of which is labeled with an element chosen from the set $\{a, b, c, d\}$. (Redundant arrows, such as an arrow from a to d, are usually omitted from such diagrams. In other words, the diagram is used as an abbreviation for its transitive closure.)



When this pomset is interpreted as a schedule, the four labels represent four *actions* and the five elements represent five *events*, each of which is a single occurrence of one of the actions. The schedule specifies that one occurrence of action a must happen before any other action. Then actions b and c may happen. Action d may happen after c has happened, and a second occurrence of action a occurs after both b and c.

The interpretation as schedules suggests a number of ways to combine several pomsets to create another. Examples are concurrence (running two schedules in parallel) and concatenation (all events of one schedule are constrained to happen before all events of the other). Formal definition of these operations leads to an algebra of pomsets.

An important goal of this thesis is to extend pomsets and their algebra to more complex schedules, such as schedules in which one may specify allowable delays between events, rather than merely the order in which events occur. For example, delay specifications might be added to the previous diagram as follows:



One interpretation, though not the only possibility, is that this diagram specifies that b follows a after a delay of at least 1 second and c follows a after at least 4 seconds, and so forth.

It would be natural to use a structure similar to metric spaces for such specifications. However, the usual definition of metric space cannot be adopted wholesale. For example, the symmetry axiom $d(a,b) = d(b,a)$ must clearly be abandoned because the delay from a to b will hardly ever be equal to the delay from b to a. Furthermore, it would be very convenient to allow distances to be other than real numbers because many different structures can then be treated within the same overall framework. For example, partially ordered sets can be seen as "metric spaces" in which the distance function is two-valued: $d(a,b) = 1$ corresponds to $a \leq b$ and $d(a,b) = 0$ to a and b unrelated.

Using this generalization other models of processes can also be treated as measured sets. For example, Gaifman and Pratt [20] allow for more complex relationships between events in that event a might precede event b either *weakly*, meaning that a and b may actually be simultaneous, or *strongly*, meaning that a must occur strictly before b. Note that if a weakly precedes b, which strongly precedes c, then a must strongly precede c. Thus a weak constraint composed with a strong constraint (in either order) yields a strong constraint. Gaifman later proposed another variant where the relation "*a causes b*" is distinct from "*a* happens to precede *b* but does not cause it" [19]. In this second model a strong constraint composed with a weak one yields a *weak* constraint. Both these models can be seen as measured set models in which the distance function is 3-valued. The difference between the two corresponds to different algebraic structures on the space of distances.

Moreover, the algebraic operations applicable to partially ordered sets of events have analogues in the more complex cases.

Such generalized metric spaces will be called *measured classes* (*measured sets* when the class is actually a set). A schedule is treated as a *labeled measured set*, which is simply a measured set, a set of labels called the *alphabet* (thought of as the set of possible actions) and an assignment of one label to each point of the measured set. A measured class whose distance function takes values in a domain D will be called a *class (or set) measured in D*.

In addition I define an algebra of labeled or unlabeled measured sets. Its operations are defined uniformly over a wide class of possible measurements and model concurrent composition of schedules, concatenation etcetera.

Formal definitions will be given in chapter 2; in the current chapter this generalization of metric space will be presented informally.

1.2 Algebra of labeled measured sets

This section is an informal description of the operations which are useful in an algebra of labeled measured sets. They will be called the *normal* operations. Each operation is illustrated using certain labeled measured sets whose measurements are taken from the positive reals together with $-\infty$ and ∞. Such structures can be considered as schedules by taking $d(a,b)$ to be the minimum delay from a to b. If b need not follow a at all, define $d(a,b) = -\infty$. The subclass of such structures in which measurements are restricted to the two values ∞ and $-\infty$ can be identified with the class of pomsets by considering $a \leq b$ to be the same as $d(a,b) = \infty$. Each of the examples given below can be converted to a pomset example by replacing each positive measurement by ∞.

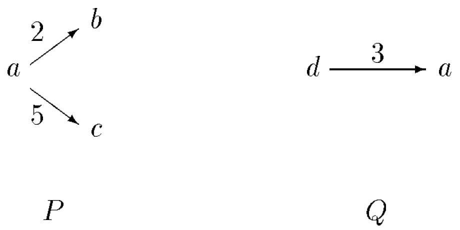


Figure 1.1: Two labeled measured sets

Concurrence. Denoted $P \| Q$. P and Q are juxtaposed with no constraints (that is, $-\infty$ constraints) between an element of either of the sets and an element of the other. This corresponds to running two schedules independently in parallel. The alphabet of $P \| Q$ is the disjoint union of the alphabet of P and the alphabet of Q, so that the two occurrences of action a are distinguished in the example of $P \| Q$ in the figure. For this reason concurrence is sometimes called *disjoint concurrence.* Taking the ordinary union instead of disjoint union gives rise to *union concurrence.*

Concatenation. Denoted $P;^x Q$. The elements of $P;^x Q$ are also the disjoint union of the elements of P and Q, but to the constraints of $P \| Q$ are added a constraint of at least x from each event of P to each event of Q. Note that $;^{-\infty}$ is identical to concurrence, and that $;^\infty$ is the usual concatenation of pomsets. Disjoint and union concatenation are defined similarly to disjoint and union concurrence. In the figure the distance from a_1 to a_2 is 12.

Orthocurrence. Denoted $P \otimes Q$. Each element is an ordered pair consisting of one element from P and one from Q. Measurements are defined by $d(\langle a,b\rangle, \langle a',b'\rangle) = d(a,a') + d(b,b')$, with $\infty + -\infty$ defined to be $-\infty$. A typical use of this operation is when P specifies messages arriving at a channel and Q is the schedule of transmission to stations along the channel. If there is a 1-second delay between the arrival of message 1 at the input and the arrival of

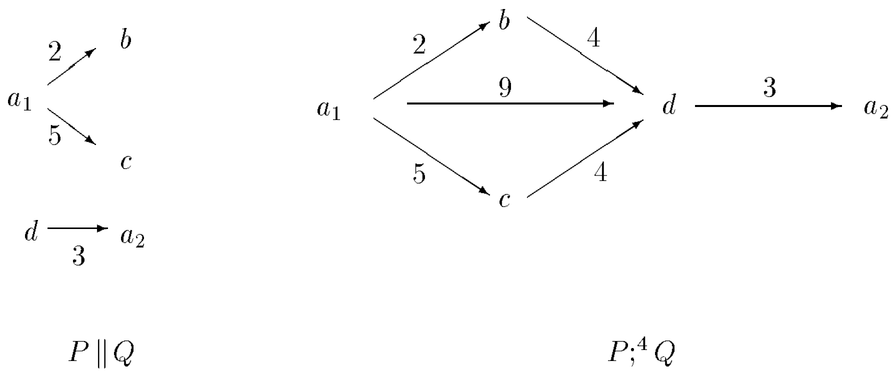


Figure 1.2: Concurrence and concatenation

message 2, and the channel has at least .5 seconds transmission time, then there will be a minimum delay of 1.5 seconds between arrival of message 1 and departure of message 2.

Pairing. Denoted $P \times Q$. Each element is an ordered pair, as in orthocurrence, but the measurement from $\langle a, b\rangle$ to $\langle a', b'\rangle$ is the *minimum* of the measurement from a to a' and the measurement from b to b'. Note that pairing is the same as orthocurrence when dealing with pomsets.

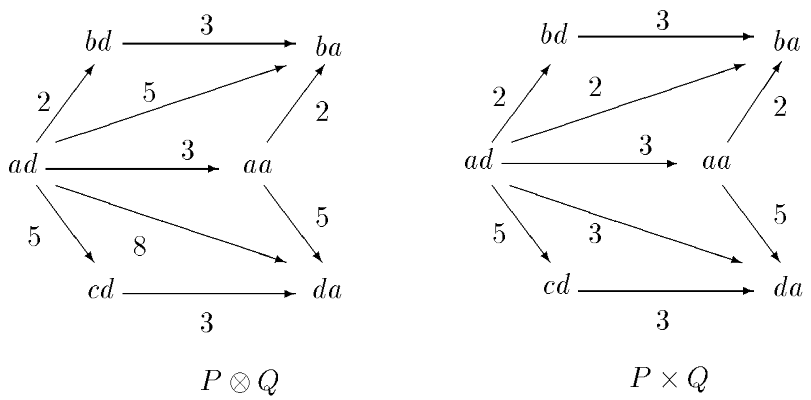


Figure 1.3: Orthocurrence and product

Standard Constants. The empty measured set, denoted **0**, is the identity for concurrence and concatenation. Two different measured singletons are of interest. One is denoted $\mathcal{I}$, and the distance from the single element $*$ to itself is 0. The other is denoted **1** and defined by $d(*, *) = \infty$. $\mathcal{I}$ is the identity for othocurrence and **1** is the identity for pairing. If the only distances are ∞ and $-\infty$ then ∞ is an identity for addition. Thus, for pomsets, $\mathcal{I} = \mathbf{1}$.

Exponentiation. Denoted $[Q, P]$. The elements of this measured set are the distance-preserving functions from Q to P, where f preserves distances if $d(fa, fb) \geq d(a, b)$ for all a and b in Q. (Observe that such functions between posets are precisely the monotone functions.) Define the distance from f to g to be the infimum of the distances from fa to ga taken over the elements of Q. When Q and P are schedules $[Q, P]$ can be interpreted as being derived from P by redefining an event to be a *pattern* of events defined by Q. In the running example there is only one distance-preserving function from Q to P, the one taking the a event to c and the d event to a. Thus $[Q, P] = \mathcal{I}$. In the corresponding pomset example, however, $[Q, P]$ has five elements. In the figure the label ab indicates the function $d \mapsto a$ and $a \mapsto b$.

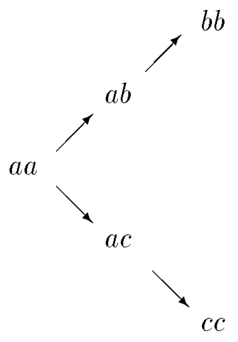


Figure 1.4: Exponentiation $[Q, P]$, pomset case only

The next set of operations rely on a concept of the *location* where an action occurs. This arises, for example, in a multiprocessor system, where each action of the system is actually performed by one of its components. Think of the components as being physically separated, so that each possible action of the whole system can be associated with the location where it could occur. A formal model of this situation is to specify a set of locations and a map to it from the alphabet of actions. Two actions that are assigned the same location are called *colocated*, and two events are said to be colocated if their actions are colocated. For the purpose of the examples, we will assume that only b and d are colocated (and, of course, that each action is colocated with itself).

Local concatenation, denoted $P\underline{;}^{x}Q$, differs from concatenation in that additional constraints are introduced only between colocated elements rather than between all of them.

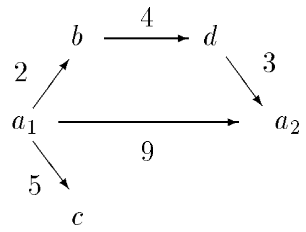


Figure 1.5: Local concatenation: $P\underline{;}^{4}Q$

Local pairing, denoted $P\underline{\times}Q$ is that subset of the pairing containing only pairs of colocated events. For the running example $P\underline{\times}Q$ has two elements labeled bd and aa and no constraints.

Fusion is a method of glueing measured sets together along a common subset. The data for fusion are two or more distance-preserving functions with the same domain, say $f\colon P \to Q$ and $g\colon P \to R$. The result is a measured set with the elements of Q and R except that the element fa is identified with ga for each a in P. In the following example think of Q as the transmission of 3 messages (a, b and a second a) from location 1 to location 2. R is the transmission of 2 a messages from 2 to 3. P is the forwarding of the a messages at location 2. The result is the processing of all messages through the system. Distances are not particularly important here and have been omitted from the figure.

1.3 Other interpretations of measured sets

Many control structures and data types other than schedules can be viewed as measured classes or labeled measured classes. An automaton is one example — the states are the elements of the class and the transitions are the measurements between them. (Start and finish states are not considered here.)

$$\begin{tikzcd} a1 \arrow[r] \arrow[d] & b1 \arrow[r] \arrow[d] & a1 \arrow[d] \\ a2 \arrow[r] & b2 \arrow[r] & a2 \end{tikzcd}$$

Q

$$\begin{tikzcd} a2 \arrow[r] \arrow[d] & a2 \arrow[d] \\ a3 \arrow[r] & a3 \end{tikzcd}$$

R

$$\begin{tikzcd} a1 \arrow[r] \arrow[d] & b1 \arrow[r] \arrow[d] & a1 \arrow[d] \\ a2 \arrow[r] \arrow[d] & b2 \arrow[r] & a2 \arrow[d] \\ a3 \arrow[rr] & & a3 \end{tikzcd}$$

Fusion (at $a2$ events)

Figure 1.6: Fusion of Q and R at $a2$ events

In fact, even when attention is confined to the simple types of measurement considered in the previous section there are two possible views of a measured set. In that section I frequently took the *conjunctive* view, where a set is a schedule. Its elements are events (*all* of which must happen, hence "conjunctive"), and the measurements require execution of the events to be followed by executions of related events. In the *disjunctive* view, the measured set is like an automaton: its elements are states, and the measurements allow passage from one state to others.

Furthermore, many other data types can be construed as classes of labeled measured sets. The class of sets is the class of measured sets when there is only one possible value for a measurement. A category is a measured class in which the measurements are sets, that is $d(a, b)$ is the set of arrows from a to b. An order enriched category (see Wand [48]) is a measured class where the measurements are partially ordered sets rather than unstructured sets. As noted above a metric space is a class where measurements are positive real numbers. Labeling adds some more possibilities. One example is multisets, which can be seen as labeled sets, although a single multiset corresponds to many isomorphic labeled sets.

Consider how some of the operations defined with concurrency in mind may be interpreted on data types.

Concurrence is disjoint union of data types. For automata concurrence corresponds to nondeterministic choice between two automata. (This view works best if automata have a *set* of start states and the start states of the result defined to be the disjoint union of the original sets of start states.)

On data types *orthocurrence* and *pairing* are two product operations. For automata both describe how two automata can be run in parallel, with each state of the combined automaton representing a state of each of the components. Either orthocurrence or pairing may be appropriate depending on the intended interpretation. If the measurement between state a and state b describes the resources required to move from a to b, then orthocurrence is the better description of their parallel composition. If, on the other hand, the measurement specifies a maximum permissible transition time between states then pairing would be a more useful operation.

Concatenation on data types is similar to disjoint union. for automata $P;Q$ represents an automaton which may begin behaving like P but escape to behaving like Q at any time. Local concatenation is similar but restricts the states which can escape to Q.

Exponentiation is easily interpreted for data types: it creates function spaces. It does not appear to have a straightforward interpretation for automata.

1.4 History and related work

The use of partial orders as a model for concurrency arose in several different ways.

One thread originates with Kahn's model of communicating processes [27], which used histories of each channel in a network to record its state. Attempts to extend this model to nondeterministic processes uncovered an anomaly [9] and Brock and Ackerman proposed adding precedence information between histories to resolve it. Pratt [40, 41, 42] gradually abandoned the association of each event with a channel, leaving only a partial order of events. He also emphasized the algebra of partial orders, and this algebra was investigated by Gischer [22], although similar notions were proposed by Birkhoff as early as 1937 [7, 8]. The extensions by Gaifman and Pratt [20], and Gaifman [19] are of particular interest since they arise very naturally as measured sets.

A second thread comes from the theory of Petri nets (see, for example, Reisig's introduction to the subject [44]) where partial orders have often been advocated for describing how one event enables another. Grabowski's use of labeled partial orders [24] is part of this thread, as is Winskel's event structures model [50, 39].

Other models based on partial orders are Degano and Montanari's Concurrent Histories [15, 16] and actor systems [1].

Of course, several of the best-known approaches to concurrency treat the concurrent execution of two actions as their sequential execution in an unspecified order. These models therefore deal with strings of events rather than partially ordered sets. Examples are Milner's CCS [36, 37] and CSP, due to Brooks, Hoare and Roscoe [11, 10, 25], and the temporal logic system of Manna and Pnueli [33]. Two ways to adapt the models presented here to such total order models are possible. The simpler of the two would just restrict attention to the total orders in the categories of partially-ordered sets and its analogues. More interesting though, are sets measured in the real numbers (or the integers) with the trivial ordering (see section 2.3 below). Such measured sets can be seen as linearly ordered, possibly with "gaps" in the order. However, this solves only part of the problem, for the system presented here does not provide a means to represent the internal decisions of a process by which it may refuse to perform an action. Perhaps one solution would be to introduce "internal" actions to represent decisions, a model similar to that discussed by Montanari and Simonelli [38].

The essential mathematics of this thesis is the theory of monoidal and enriched categories, used in a way suggested by Lawvere [30]. Recently several researchers have applied some of the same theory, but it must be stressed that they use it in a rather different way.

Main and Benson [4, 5, 3, 32], for example, use free modules over semirings to model the inputs and outputs of a nondeterministic computation. Tensor products (hence monoidal categories) arise naturally here when dealing with multiple inputs or outputs.

Similarly, Degano, Meseguer and Montanari treat multisets as free modules over the semiring of the natural numbers to give an algebraic treatment of Petri nets [35, 14]. In this case arrows of the category represent transitions, and tensor product corresponds to running a number of transitions concurrently. (In the system presented in this thesis the concurrence operation turns out to be categorical sum, not tensor product.) Meseguer and Martì-Oliet [34] show that Girard's linear logic [21], Petri nets and monoidal categories are all related, but while this work certainly links concurrency and monoidal categories in a very interesting way, no deep connection between it and this thesis is known.

1.5 Summary of following chapters

In Chapter 2, I define the MCE-categories to be those that are symmetric monoidal, closed, complete and cocomplete. Such categories possess those properties needed to define all the normal operations. I define the normal operations in this setting and show that the category of pomsets is one such category with operations given by the general definition. A wider class of categories, called the MC-categories, lacking only the closure property, is also of interest.

But proving any given category to be MC or MCE may be somewhat tedious, and many of the categories of interest can be constructed from simpler categories by combinations of two constructions called *enrichment* and *labeling*. In chapters 3 and 4 a number of results are proved to show the existence of (co)limits in the constructed categories when given related properties of the simpler categories from which they were constructed. Similar results show closure of the constructed categories under appropriate conditions.

The result can be used as an algebra of schedules in which the notion of time has become a parameter rather than an unchangeable fundamental. One can choose whatever notion of time is appropriate to a particular application and, when expressed as a suitable monoidal category, this choice generates a corresponding category of schedules. Operations for combining schedules are defined uniformly and independently of the choice of time, and are automatically interpreted in a way suitable for that choice.

As regards prerequisites, I assume that the reader is familiar with the notions of functor, natural transformation, limit, colimit, adjunction and comma category at the level to be found in any introduction to category theory, such as Mac Lane's [31]. The theory of 2-categories is also used at some points. Mac Lane treats these briefly, more information can be found in a paper by Kelly and Street [28].

As regards notation: function and functor application are usually denoted without parentheses (that is, FX rather than $F(X)$). The composite arrow $A \xrightarrow{f} B \xrightarrow{g} C$ is denoted gf (not $g \circ f$ or fg). Subscripts denoting the components of natural transformations are omitted whenever they can be derived from the context. So if $\nu\colon F \to G$ is a natural transformation, the component $\nu_X\colon FX \to GX$ will often be written simply as $\nu\colon FX \to GX$. Identity arrows are denoted by 1.

Chapter 2

MC Categories

In this chapter I define the normal operations formally, using a framework provided by the theory of monoidal categories. I define the MC-categories, in which the normal operations other than exponentiation are well-defined, and the MCE-categories, in which all the normal operations are defined. The MC-categories include the categories of measured classes (along with other possibilities). I show that the category of pomsets is MC and that the abstract definitions yield the expected operations (sometimes with minor variants) in this case.

2.1 Monoidal categories

Roughly speaking, a monoidal category is a category equipped with a multiplication (that is, a functor of two arguments) called *tensor product* and denoted by $\otimes$, and an identity element denoted by I. Such categories play two roles in the theory described here. Firstly, they serve as the domains of distance functions of measured sets. In this case each object of the monoidal category represents a distance and tensor product represents addition of distances. The arrows of the category represent certain relations between distances that arise when distances are seen as constraints. For simple cases, in which the category is a partial order, $A \leq B$ means that A is a weaker constraint than B in the sense that two events satisfying constraint B will also satisfy A. For example, if constraints are lower bounds on distances then two events separated by at least 5 units will certainly be separated by as least 3 units, so $3 \leq 5$.

The second role of monoidal categories arises because the class of sets measured in a given domain forms a monoidal category, in a manner to be described below. It is convenient to define the normal operations purely in terms of this monoidal category structure without specific reference to the fact that the objects are measured sets. In this case the tensor product is the orthocurrence operation described in section 1.2.

The two roles are related because given a space of measurements viewed as monoidal category, a standard construction yields its category of measured sets.

The following definition of monoidal category is standard (Mac Lane [31] and Arbib and Manes [2] treat them briefly; detailed treatments are by Eilenberg and Kelly [17] and Kelly [29]). The usual associativity and identity axioms of a monoid are weakened by requiring only that $(A \otimes B) \otimes C$ be *isomorphic* to $A \otimes (B \otimes C)$ rather than that the two expressions be equal. Similar isomorphisms replace the standard left and right identity laws. In order to lift results from the identity case to the isomorphism case, though, it is necessary that these isomorphism be natural, and that compositions of them satisfy certain conditions called "coherence." (Note that if α, λ and ρ are all identity maps then naturality and coherence hold trivially.) If we insisted on identities rather than isomorphisms many important examples would fail to satisfy the definition, including the categories of measured sets mentioned in the introduction to this chapter.

Definition 2.1 A *monoidal category* $\mathcal{D} = \langle \mathcal{D}_0, \otimes, I, \alpha, \lambda, \rho \rangle$ consists of

i) A category $\mathcal{D}_0$;

ii) A functor $\otimes : \mathcal{D}_0 \times D_0 \to \mathcal{D}_0$ (usually written in infix notation);

iii) An object I of $\mathcal{D}$;

iv) An isomorphism $\alpha : (A \otimes B) \otimes C \cong A \otimes (B \otimes C)$ natural in A, B and C;

v) An isomorphism $\lambda : I \otimes A \cong A$ natural in A; and

vi) An isomorphism $\rho : A \otimes I \cong A$ natural in A.

These data must satisfy certain *coherence conditions*, namely the commutativity of the following two diagrams.

$$\begin{tikzcd}
((A \otimes B) \otimes C) \otimes D \arrow[r, "\alpha"] \arrow[d, "\alpha \otimes 1"'] & (A \otimes B) \otimes (C \otimes D) \arrow[r, "\alpha"] & A \otimes (B \otimes (C \otimes D)) \\
(A \otimes (B \otimes C)) \otimes D \arrow[rr, "\alpha"] & & A \otimes ((B \otimes C) \otimes D) \arrow[u, "1 \otimes \alpha"']
\end{tikzcd}$$

$$\begin{tikzcd}
(A \otimes I) \otimes B \arrow[rr, "\alpha"] \arrow[dr, "\rho \otimes 1"'] & & A \otimes (I \otimes B) \arrow[dl, "1 \otimes \lambda"] \\
& A \otimes B &
\end{tikzcd}$$

∎

The coherence laws imply that any two parallel arrows constructed solely by composition, $\otimes$ and taking inverses from α, λ, ρ and identity arrows must be equal [31].

The next definition is for a *symmetric* monoidal category, which would be a commutative monoid in the category of categories except that, once again, the monoid laws need hold only up to the specified natural isomorphisms. In the interpretation of a monoidal category as a space of measurements symmetry specifies that addition of distances must be commutative. Almost all the monoidal categories we encounter will be symmetric.

Definition 2.2 A *symmetric monoidal category* $\mathcal{D} = \langle \mathcal{D}_0, \otimes, I, \alpha, \lambda, \rho, \gamma \rangle$ consists of a monoidal category, $\langle \mathcal{D}_0, \otimes, I, \alpha, \lambda, \rho \rangle$, together with an isomorphism $\gamma : A \otimes B \cong B \otimes A$ natural in both A and B and satisfying the following laws:

i) γ is an involution:

$$\begin{tikzcd}
A \otimes B \arrow[rr, "1"] \arrow[dr, "\gamma"'] & & A \otimes B \\
& B \otimes A \arrow[ur, "\gamma"'] &
\end{tikzcd}$$

ii) $\gamma\rho = \lambda$:

$$\begin{tikzcd} I \otimes A \arrow[rr, "\gamma"] \arrow[dr, "\lambda"'] & & A \otimes I \arrow[dl, "\rho"] \\ & A & \end{tikzcd}$$

iii) γ and α harmonize:

$$\begin{tikzcd} (A \otimes B) \otimes C \arrow[r, "\alpha"] \arrow[d, "\gamma \otimes 1"'] & A \otimes (B \otimes C) \arrow[r, "\gamma"] & (B \otimes A) \otimes C \arrow[d, "\alpha"] \\ (B \otimes C) \otimes A \arrow[r, "\alpha"] & B \otimes (A \otimes C) \arrow[r, "1 \otimes \gamma"] & B \otimes (C \otimes A) \end{tikzcd}$$

∎

2.2 Enriched categories

Given a monoidal category $\mathcal{D}$ we may define categories *enriched in* $\mathcal{D}$, or simply $\mathcal{D}$-*categories*. They consist, in part, of a class of objects together with an assignment of an object of $\mathcal{D}$ to each ordered pair of objects. When $\mathcal{D}$ is interpreted as a space of measurements, a $\mathcal{D}$-category can be seen as a measured class: the objects of the $\mathcal{D}$-category are the elements of the measured class and the object of $\mathcal{D}$ assigned to a pair is the distance between them. Further conditions of the definition correspond to a triangle inequality on measurements and a requirement that the distance from an element to itself is not arbitrary.

In addition to the preceding motivation, it should also be observed that **Set**, the category of sets, can be considered a monoidal category by defining $\otimes$ to be cartesian product and I to be a singleton set. For the case $\mathcal{D} = \mathbf{Set}$ the following definition is just an alternative definition of a category $\mathcal{A}$ in which $\mathcal{A}(A, B)$ is the set of arrows from A to B, $\mu\colon \mathcal{A}(A, B) \times \mathcal{A}(B, C) \to \mathcal{A}(A, C)$ is the composition map $\mu(f, g) = gf$ and $\iota\colon 1 \to \mathcal{A}(A, A)$ is the constant map to the identity arrow on A. The commutativity requirements are just the statements that composition is associative and composition with identity arrows is an identity.

Definition 2.3 Let $\mathcal{D} = \langle \mathcal{D}_0, \otimes, I, \alpha, \lambda, \rho \rangle$ be a monoidal category. A *category enriched in* $\mathcal{D}$ is defined by:

i) A class $|\mathcal{A}|$ called the *objects* of $\mathcal{A}$; and

ii) For each pair of objects A and B in $|\mathcal{A}|$ an object $\mathcal{A}(A, B)$; and

iii) For each triple of objects A, B and C in $|\mathcal{A}|$, a morphism of $\mathcal{D}$

$$\mu_{ABC}\colon \mathcal{A}(A, B) \otimes \mathcal{A}(B, C) \to \mathcal{A}(A, C); \text{ and}$$

iv) For each object A of $\mathcal{A}$ a morphism of $\mathcal{D}$ $\iota_A\colon I \to \mathcal{A}(A, A)$.

These data must satisfy the following conditions:

i) μ is associative:

$$\begin{tikzcd}
(\mathcal{A}(A,B) \otimes \mathcal{A}(B,C)) \otimes \mathcal{A}(C,D) \arrow[rr, "\alpha"] \arrow[d, "\mu \otimes 1"'] & & \mathcal{A}(A,B) \otimes (\mathcal{A}(B,C) \otimes \mathcal{A}(C,D)) \arrow[d, "1 \otimes \mu"] \\
\mathcal{A}(A,C) \otimes \mathcal{A}(C,D) \arrow[dr, "\mu"'] & & \mathcal{A}(A,B) \otimes \mathcal{A}(B,D) \arrow[dl, "\mu"] \\
& \mathcal{A}(A,D) &
\end{tikzcd}$$

ii) ι is a left identity

$$\begin{tikzcd}
I \otimes \mathcal{A}(A,B) \arrow[rr, "\iota \otimes 1"] \arrow[dr, "\lambda"'] & & \mathcal{A}(A,A) \otimes \mathcal{A}(A,B) \arrow[dl, "\mu"] \\
& \mathcal{A}(A,B) &
\end{tikzcd}$$

iii) ι is a right identity

$$\begin{tikzcd}
\mathcal{A}(A,B)\otimes I \arrow[r, "1\otimes\iota"] \arrow[dr, "\rho"'] & \mathcal{A}(A,B)\otimes\mathcal{A}(B,B) \arrow[d, "\mu"] \\
 & \mathcal{A}(A,B)
\end{tikzcd}$$

If the class $|\mathcal{A}|$ is a set then $\mathcal{A}$ is called a *small* $\mathcal{D}$-category. ∎

Finally, I define an important property that many of the examples of monoidal categories possess.

Definition 2.4 A monoidal category $\mathcal{D}$ is *closed* if for each object A of $\mathcal{D}_0$ the functor

$$-\otimes A\colon \mathcal{D}_0 \to \mathcal{D}_0$$

has a right adjoint. ∎

Note that if $\mathcal{D}$ is symmetric (as will usually be the case in our examples) then closure is equivalent to the existence of a right adjoint for either $-\otimes A$ or $A\otimes -$.

In this case there exists a unique functor $\mathcal{D}_0^{\mathrm{op}}\times\mathcal{D}_0\to\mathcal{D}_0$, denoted by

$$[-,-]\colon \mathcal{D}_0^{\mathrm{op}}\times\mathcal{D}_0\to\mathcal{D}_0,$$

such that for each A, $[A,-]$ is right adjoint to $A\otimes -$ [31, page 100]. This functor is called the *internal-hom* of the closed monoidal category $\mathcal{D}$. (An alternative name for this functor is *exponentiation* and an alternative notation for $[A,B]$ is B^A. I use the second notation only to refer to ordinary functor categories.)

A closed $\mathcal{D}$ may be considered as a category enriched in itself, by defining $\mathcal{D}(A,B)$ to be $[A,B]$. μ and ι can then be defined in terms of the adjunction between $\otimes$ and $[-,-]$.

2.3 Examples

Here are some important examples of monoidal categories and the categories enriched in them.

i) Any category having finite products can be given a symmetric monoidal structure by defining $\otimes$ taken to be categorical product, I to be a terminal object, and defining α,

ρ, λ and γ using the universal property of product. This structure is called a *cartesian monoidal category*, and if it is closed then it is *cartesian closed*.

The category **Set** has already been mentioned. Another important example is **Cat** the category of small categories. A **Cat**-category is also known as a 2-category. Mac Lane [31] gives an alternative definition of 2-category in terms of objects, arrows and 2-cells. Kelly and Street [28] state that the two definitions are equivalent. Both **Set** and **Cat** are closed, and hence cartesian closed.

Note that many such examples would be excluded if α, λ, ρ and γ were required to be identities. Cartesian product of sets, for example, is *not* associative. $(A \times B) \times C$ is merely isomorphic to $A \times (B \times C)$ in an obvious way, and the necessary properties of this isomorphism are captured in the coherence axioms.

ii) The category $\mathbf{Mod}_R$ of modules over a commutative ring R is monoidal with $\otimes$ being the usual tensor product of such modules, I being R considered as an R-module. In particular the category of small abelian groups **Ab** is such a category — take R to be $\mathbf{Z}$. **Ab**-categories are sometimes called preadditive categories. **Ab** is closed, but not cartesian.

Examples like **Set**, **Cat** and **Ab** are historically important to the development of enriched category theory, but in applications to computer science other examples are of more interest.

i) The ordinal **2** considered as a cartesian closed category. This can be seen as the category of boolean truth values, with product being conjunction. **2**-categories are preorders: the composition law reduces to the statement of transitivity and the identity law to that of reflexivity. In order to avoid confusion with 2-categories I will call such categories **Bool**-categories.

ii) The nonnegative real numbers with the usual ordering and tensor product defined to be ordinary addition. This monoidal category is not closed, for if tensor were a left adjoint it would preserve all colimits. In particular, it would preserve the initial object 0, but in fact $x + 0 \neq 0$ in general. However the category can be made closed by adjoining $-\infty$ as a new object with the property $x + (-\infty) = -\infty$ for all x. An ∞ object is then required as the value of $[-\infty, 0]$. It has the property $x + \infty = \infty$ for all x except $x = -\infty$. This category (including both ∞ and $-\infty$) will be denoted by $\mathbf{R}_+$.

The internal-hom of $\mathbf{R}_+$ is

$$[x, y] = \begin{cases} 0 & \text{if } x = y = -\infty \\ y - x & \text{if } y \geq x \\ -\infty & \text{if } y < x. \end{cases}$$

Categories enriched over $\mathbf{R}_+$ can be seen as schedules of events with minimum delays specified between each pair of events. The composition law is consistent with this interpretation; it states that the minimimum delay from A to B must be at least as great as the minimum delay from a to c plus the minimum delay from c to b, for any event c. The identity law says that $d(a, a) \geq 0$ for all a and then the composition law implies that if $d(a, a) > 0$ then $d(a, a) = \infty$. Further application of the composition law shows that for such an a *all* distances $d(a, b)$ and $d(b, a)$ are infinite.

iii) (due to W. Lawvere [30]) The nonnegative real numbers with the opposite of the usual order, and tensor product being addition is closed with internal-hom defined by truncated subtraction.

$$[x, y] = \begin{cases} y - x & \text{if } y \geq x \\ 0 & \text{otherwise.} \end{cases}$$

However it is not cocomplete. To make it cocomplete (while preserving closedness), include ∞ as an initial object. This category is denoted $\mathbf{R}_+^{\mathrm{op}}$.

$\mathbf{R}_+^{\mathrm{op}}$-categories are similar to metric spaces, with $d(a, b)$ being the distance from a to b. The identity law reduces to the statement that $d(a, a) = 0$ and the composition law to the triangle inequality. $\mathbf{R}_+^{\mathrm{op}}$-categories need not satisfy the law $d(a, b) = 0$ implies $a = b$ nor the symmetry law.

$\mathbf{R}_+^{\mathrm{op}}$-categories can be seen as schedules of events with maximum delays between each pair of events.

Pratt [43] has observed that this category has appeared in computer science in the guise of a *semiring* of distances, in which semiring addition is **min** and semiring multiplication is addition of reals. In categorical terms, addition is coproduct, multiplication is tensor product, and distributivity is ensured by closure. The ordinal **2** has also been treated the same way. Treating these sets as semirings allowed Robert and

Ferland [45] to treat Roy [46] and Warshall's [49] algorithm for transitive closure of a relation, and Floyd's [18] shortest-path algorithm, as a single algorithm on matrices over different semirings.

Pratt also notes that Kleene's algorithm for closing an automaton is another example, in which the underlying category contains languages over a given alphabet, ordered by inclusion and with (non-symmetric) product being concatenation.

In the current setting a matrix over $\mathcal{D}$ is a "$\mathcal{D}$-graph," and Robert and Ferland's algorithm is precisely the construction of a free $\mathcal{D}$-category on the $\mathcal{D}$-graph.

iv) The category of sets of nonnegative real numbers ordered by inclusion is a monoidal closed category under the tensor product

$$X \otimes Y = \{\, x + y \mid x \in X \text{ and } y \in Y \,\}.$$

denoted by **SR**. The internal-hom is given by

$$[X, Y] = \{\, z \mid z + X \subseteq Y \,\}.$$

Both $\mathbf{R}_+$ and $\mathbf{R}_+^{\mathrm{op}}$ are sub-monoidal-categories of **SR**. $\mathbf{R}_+$ consists of those intervals of the form $[0, x)$ for $0 \leq x \leq \infty$. ∞ corresponds to the set of all nonnegative reals and $-\infty$ to the empty set. $\mathbf{R}_+^{\mathrm{op}}$ consists of the intervals $[x, \infty)$. One can also define other interesting subcategories of **SR**. For example we could consider only the closed intervals rather than arbitrary subsets.

SR-categories can be seen as schedules of events where $d(a, b)$ is a set of possible delays from a to b.

v) Each example involving the real numbers has a corresponding example where the nonnegative reals are replaced by the natural numbers.

vi) Any monoid can be made into a monoidal category by considering it as a discrete category. For such a category to be closed it must be a group. An example is the set of real numbers $\mathbf{R}$ with the discrete ordering and tensor product being addition. A category enriched in $\mathbf{R}$ is like a schedule with exact delays specified between each event; the time of execution of any single event determines the time of every other event in the schedule.

vii) A simple monoidal category arises from the Gaifman-Pratt *prosset* model of concurrency [20] in which the set of events is equipped with two relations: a preorder $\leq$ and an irreflexive partial order $<$ with the condition that $a < b$ implies $a \leq b$. The intention is that $a \leq b$ be interpreted as "a happens no later than b" and $a < b$ as "a happens strictly earlier than b."

This model can be fitted into our framework by taking $\mathcal{D}$ to be the ordinal number 3, with commutative tensor product defined by $0 \otimes x = 0$; $1 \otimes x = x$ and $2 \otimes 2 = 2$. This category is denoted $\mathbf{3}'$. Then we can identify a prosset with a category enriched over $\mathbf{3}'$, by taking $d(x, y) = 2$ to mean $x < y$ and $d(x, y) = 1$ to mean $(x \leq y) \wedge (x \not< y)$. Here the composition law becomes the statement that $a < b < c$ implies $a < c$ and $a < b \leq c$ or $a \leq b < c$ or $a \leq b \leq c$ all imply $a \leq c$.

Gaifman has also introduced another definition of processes in which there are two orderings: $<^c$ for causal relationships and $<^t$ for accidental temporal relationships. Here $a <^c b$ implies $a <^t b$ but not the converse. $a <^c b$ and $b <^t c$ together imply $a <^t c$.

Schedules with such relationships can be seen as $\mathbf{3}$-categories, where now the ordinal $\mathbf{3}$ is considered as a cartesian monoidal category. For this tensor product $0 \otimes x = 0$ $2 \otimes x = x$ and $1 \otimes 1 = 1$. We can identify $a <^c b$ with $d(a, b) = 2$ and $d(a, b) = 1$ with $(a <^t b) \wedge (a \not<^c b)$. The composition law for $\mathbf{3}$-categories now becomes precisely the statement that successive temporal constraints imply a temporal constraint (since $1 \otimes 1 = 1$) and that a temporal constraint composed with a causal constraint imply a temporal constraint (since $1 \otimes 2 = 1$)

Both $\mathbf{3}$ and $\mathbf{3}'$ are special cases of a general way of representing multiple-partial-order models of concurrency. These models correspond to nonzero finite ordinals considered as categories and equipped with a symmetric idempotent tensor product. The ordering on elements of the ordinal correspond to implications between the partial order relations. The tensor product defines how compositions of different partial orders are to be interpreted; symmetry is the requirement that the order of composition is not relevant to the result and idempotence is required to show that each element does, in fact, correspond to a partial order.

Monoidal categories with all the properties described in the previous paragraph have a rather interesting structure. Firstly, note that one can define a new partial ordering $\prec$ on the elements by $a \prec b$ if and only if $a \otimes b = a$. Symmetry ensures that $\prec$ is reflexive and $a \prec b \prec c$ iff $a \otimes b = a$ and $b \otimes c = b$, whence $a \otimes c = (a \otimes b) \otimes c = a \otimes (b \otimes c) = a \otimes b = a$. So $\prec$ is transitive. It is obvious that $\prec$ is antisymmetric. Further properties of $\prec$ are given by the following.

Proposition 2.5 *(due to V. Pratt.)*

i) $\prec$ *is a total order;*

ii) For any a *and* b $a \otimes b$ *is the minimum of* a *and* b *under* $\prec$*;*

iii) When elements are enumerated in $\leq$*-order they first ascend in the* $\prec$*-order until* I *is reached, then descend in the* $\prec$*-order;*

iv) The ordinal is a closed monoidal category if and only if 0 *is minimal under* $\prec$*.*

Proof:

i) Without loss of generality assume that $a \leq b$. If $I \leq a \otimes b$ then $b = I \otimes b \leq a \otimes b \otimes b = a \otimes b \leq b \otimes b = b$. Hence $a \otimes b = b$. If $I \not\leq a \otimes b$ then $a \otimes b \leq I$, in which case $a = a \otimes a \leq a \otimes b = a \otimes a \otimes b \leq a \otimes I = a$, so $a \otimes b = a$.

ii) $a \otimes a \otimes b = a \otimes b$ (by idempotence) so $a \otimes b \prec a$ and similarly $a \otimes b \prec b$. But by part (i) either $a \otimes b = a$ or $a \otimes b = b$, so $a \otimes b$ must be the smaller of a and b under the $\prec$ order.

iii) By part (ii), I is maximal under $\prec$. If $a \leq b \leq I$ then $a = a \otimes a \leq a \otimes b \leq a \otimes I = a$, so $a \otimes b = a$, that is $a \prec b$. If $I \leq a \leq b$ then $b = I \otimes b \leq a \otimes b \leq b \otimes b = b$, so $b \prec a$.

iv) If the ordinal is closed then $0 \otimes a = 0$ because $- \otimes a$, being a left adjoint, must preserve initiality. Conversely, if $0 \otimes a = 0$ then for every a and b the set $\{ x \mid a \otimes x \leq b \}$ is nonempty, and by finiteness and the fact that $\leq$ is a total order it contains an element maximal under $\leq$. Defining $[a, b]$ to be this element constructs a right adjoint of $\otimes$.

∎

A consequence of this proposition is that Gaifman's two models of concurrency are the only possible two-partial-order models, as shown in the following.

Corollary 2.6 *There are precisely two idempotent closed symmetric monoidal structures on the ordinal* $\mathbf{3}$*, namely the monoidal categories* $\mathbf{3}$ *and* $\mathbf{3}'$ *defined above.*

Proof: By part (iii) of the proposition such a category is completely specified by listing the three elements in the $\prec$ order. If $I = 0$ the only possible order is 210. If $I = 2$, the only possibility is 012. For $I = 1$ both 021 and 201 are possible. By part (iv) of the proposition only two of these are closed, namely 012, which describes $\mathbf{3}$, and 021, which describes $\mathbf{3}'$. ∎

2.4 The category of $\mathcal{D}$-categories

As noted at the beginning of section 2.1, the class of $\mathcal{D}$-categories forms a category. I will now define this category.

The arrows of the category are analogous to functors. Indeed, in the case that $\mathcal{D} = \mathbf{Set}$ (so that a $\mathcal{D}$-category is an ordinary category) the following simply restates the usual definition of functor. However, it has the advantage that it also applies to other choices of $\mathcal{D}$.

Definition 2.7 Given a monoidal category $\mathcal{D}$ and $\mathcal{D}$-categories $\mathcal{A}$ and $\mathcal{B}$, a $\mathcal{D}$-functor $F: \mathcal{A} \to \mathcal{B}$ consists of the following data:

i) A map (also called F) of the objects of $\mathcal{A}$ to the objects of $\mathcal{B}$;

ii) For each pair A, B of objects of $\mathcal{A}$, an arrow in $\mathcal{D}$ $F_{AB}: \mathcal{A}(A, B) \to \mathcal{B}(FA, FB)$.

This data must make the following diagrams commute:

i) F preserves composition:

$$\begin{tikzcd}
\mathcal{A}(A,B) \otimes \mathcal{A}(B,C) \arrow[r, "F \otimes F"] \arrow[d, "\mu"'] & \mathcal{B}(FA,FB) \otimes \mathcal{B}(FB,FC) \arrow[d, "\mu"] \\
\mathcal{A}(A,C) \arrow[r, "F"] & \mathcal{B}(FA,FC)
\end{tikzcd}$$

ii) F preserves identity:

$$\begin{tikzcd} I \arrow[rr, "\iota"] \arrow[dr, "\iota"'] & & \mathcal{A}(A,A) \arrow[dl, "F"] \\ & \mathcal{B}(FA,FA) & \end{tikzcd}$$

∎

2.4.1 Examples of $\mathcal{D}$-functors

i) **Set**-functors are ordinary functors between ordinary categories. For this case the first condition means that a functor preserves composition and the second means that a functor preserves identity arrows.

ii) **Cat**-functors are precisely 2-functors.

iii) **Bool**-functors are monotone maps between preordered classes.

iv) $\mathbf{R}_+$-functors are *noncontracting* maps between schedules. That is, $F: \mathcal{A} \to \mathcal{B}$ must satisfy $\mathcal{A}(X,Y) \leq \mathcal{B}(FX,FY)$ for all objects X and Y of $\mathcal{A}$.

v) $\mathbf{R}_+^{\mathrm{op}}$-functors are *nonexpanding* maps, which are the maps satisfying

$$\mathcal{A}(X,Y) \geq \mathcal{B}(FX,FY).$$

Define the identity $\mathcal{D}$-functor on any $\mathcal{D}$-category to be the identity on both objects and homobjects.

Given $\mathcal{D}$-categories $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$, and $\mathcal{D}$-functors $F: \mathcal{A} \to \mathcal{B}$ and $G: \mathcal{B} \to \mathcal{C}$, define the composition $GF: \mathcal{A} \to \mathcal{B}$ by defining the underlying object map by

$$(GF)A = G(FA);$$

and then defining $(GF)_{AA'}: \mathcal{A}(A,A') \to \mathcal{C}(GFA, GFA')$ to be the composite

$$\mathcal{A}(A,A') \xrightarrow{F_{AA'}} \mathcal{B}(FA,FA') \xrightarrow{G_{(FA,FA')}} \mathcal{C}(GFA,GFA').$$

This composition is associative and the identity functors are in fact identities for the composition. Thus, we have the following proposition.

Proposition 2.8 *The class of small $\mathcal{D}$-categories and $\mathcal{D}$-functors is a category.*

This category will be denoted by $\mathcal{D}!$.

2.5 Tensor product of $\mathcal{D}$-categories

Not only do the $\mathcal{D}$-categories form a category but the tensor product of $\mathcal{D}$ lifts to a tensor product of $\mathcal{D}$-categories, provided that $\mathcal{D}$ is symmetric monoidal. This is important here because if $\mathcal{D}$ is a space of measurements then the lifted tensor product turns out to be exactly the orthocurrence of sets measured in $\mathcal{D}$.

So let $\mathcal{D}$ be symmetric monoidal. The tensor product of two $\mathcal{D}$-categories $\mathcal{A}$ and $\mathcal{B}$ is defined as follows (see Kelly [29]). The objects of $\mathcal{A} \otimes \mathcal{B}$ are of the form $\langle A, B\rangle$ where A is an object of $\mathcal{A}$ and B an object of $\mathcal{B}$. Homobjects are defined by

$$(\mathcal{A} \otimes \mathcal{B})(\langle A_1, B_1\rangle, \langle A_2, B_2\rangle) = \mathcal{A}(A_1, A_2) \otimes \mathcal{B}(B_1, B_2).$$

To define composition in $\mathcal{A} \otimes \mathcal{B}$ first define the interchange natural isomorphism in $\mathcal{D}$:

$$t\colon (W \otimes X) \otimes (Y \otimes Z) \cong (W \otimes Y) \otimes (X \otimes Z).$$

There are several ways to construct such a map by combining α, γ and identity maps using tensor product, inversion and composition. The coherence conditions will ensure that any two constructions of this form denote the same map but, for definiteness, define t to be the composite:

$$\begin{array}{rcl}
(W \otimes X) \otimes (Y \otimes Z) & \xrightarrow{\alpha} & W \otimes (X \otimes (Y \otimes Z)) \\
& \xrightarrow{1\otimes\alpha^{-1}} & W \otimes ((X \otimes Y) \otimes Z) \\
& \xrightarrow{1\otimes(\gamma\otimes 1)} & W \otimes ((Y \otimes X) \otimes Z) \\
& \xrightarrow{1\otimes\alpha} & W \otimes (Y \otimes (X \otimes Z)) \\
& \xrightarrow{\alpha^{-1}} & (W \otimes Y) \otimes (X \otimes Z)).
\end{array}$$

$\mu^{\mathcal{A}\otimes\mathcal{B}}$, the composition map in $\mathcal{A} \otimes \mathcal{B}$ is then defined to be the composite map:

$$\begin{array}{rl}
\multicolumn{2}{l}{(\mathcal{A} \otimes \mathcal{B})(\langle A_1, B_1\rangle, \langle A_2, B_2\rangle) \otimes (\mathcal{A} \otimes \mathcal{B})(\langle A_2, B_2\rangle, \langle A_3, B_3\rangle)} \\
= & (\mathcal{A}(A_1, A_2) \otimes \mathcal{B}(B_1, B_2)) \otimes (\mathcal{A}(A_2, A_3) \otimes \mathcal{B}(B_2, B_3))
\end{array}$$

$$
\begin{array}{cl}
\xrightarrow{t} & (\mathcal{A}(A_1, A_2) \otimes \mathcal{A}(A_2, A_3)) \otimes (\mathcal{B}(B_1, B_2) \otimes \mathcal{B}(B_2, B_3)) \\
\xrightarrow{\mu \otimes \mu} & \mathcal{A}(A_1, A_3) \otimes \mathcal{B}(B_1, B_3) \\
= & (\mathcal{A} \otimes \mathcal{B})(\langle A_1, B_1 \rangle, \langle A_3, B_3 \rangle)
\end{array}
$$

$i^{\mathcal{A}\otimes\mathcal{B}}$, the identity map is defined to be the composite arrow

$$
\begin{array}{ccl}
I & \xrightarrow{\rho} & I \otimes I \\
 & \xrightarrow{i \otimes i} & \mathcal{A}(A, A) \otimes \mathcal{B}(B, B) \\
 & = & (\mathcal{A} \otimes \mathcal{B})(\langle A, B \rangle, \langle A, B \rangle).
\end{array}
$$

Proposition 2.9 *$\mathcal{A} \otimes \mathcal{B}$, with μ and i defined above, is a $\mathcal{D}$-category.*

Proof: I show only the associativity law for $\mathcal{A} \otimes \mathcal{B}$. The identity laws are proved by a similar argument.

Let X_1, X_2, X_3 and X_4 be any objects of $\mathcal{A}$ and Y_i, for $1 \leq i \leq 4$ be any objects of $\mathcal{B}$. In the diagram on the following page $\mathcal{A}(X_i, X_{i+1})$ is abbreviated as A_i, $\mathcal{A}(X_1, X_3)$ is abbreviated as A_{13} and so on. Similarly $\mathcal{B}(Y_i, Y_{i+1})$ is abbreviated as B_i and so on.

I wish to show that the two outside edges of that diagram are equal.

But the pentagon in the bottom right corner commutes because it is a tensor product of the pentagons that express associativity in $\mathcal{A}$ and $\mathcal{B}$. The squares in the top right and bottom left corners commute because t is natural, and the hexagon in the top left corner commutes by coherence of $\mathcal{D}$.

∎

I now extend the definition of $\otimes$ to $\mathcal{D}$-functors. Suppose that $F: \mathcal{A} \to \mathcal{A}'$ and $G: \mathcal{B} \to \mathcal{B}'$ are $\mathcal{D}$-functors and that X_1 and X_2 are objects of $\mathcal{A}$ and Y_1 and Y_2 are objects of $\mathcal{B}$. Define the object map of $F \otimes G$ by

$$(F \otimes G)\langle X_1, Y_1 \rangle = \langle FX_1, GY_1 \rangle.$$

Now the arrow $F \otimes G_{\langle X_1, Y_1 \rangle \langle X_2, Y_2 \rangle}$ must be of the form

$$(\mathcal{A} \otimes \mathcal{B})(\langle X_1, Y_1 \rangle, \langle X_2, Y_2 \rangle) \to (\mathcal{A}' \otimes \mathcal{B}')((F \otimes G)\langle X_1, Y_1 \rangle, (F \otimes G)\langle X_2, Y_2 \rangle).$$

$$
\begin{tikzcd}
((A_1 \otimes B_1) \otimes (A_2 \otimes B_2)) \otimes (A_3 \otimes B_3) \arrow[r, "t \otimes 1"] \arrow[d, "\alpha"'] & ((A_1 \otimes A_2) \otimes (B_1 \otimes B_2)) \otimes (A_3 \otimes B_3) \arrow[r, "(\mu \otimes \mu) \otimes 1"] \arrow[d, "t"] & (A_{12} \otimes B_{12}) \otimes (A_3 \otimes B_3) \arrow[d, "t"] \\
(A_1 \otimes B_1) \otimes ((A_2 \otimes B_2) \otimes (A_3 \otimes B_3)) \arrow[d, "1 \otimes t"'] & ((A_1 \otimes A_2) \otimes A_3) \otimes ((B_1 \otimes B_2) \otimes B_3) \arrow[r, "(\mu \otimes 1) \otimes (\mu \otimes 1)"] \arrow[d, "\alpha \otimes \alpha"] & (A_{12} \otimes A_3) \otimes (B_{12} \otimes B_3) \arrow[dd, "\mu \otimes \mu"] \\
(A_1 \otimes B_1) \otimes ((A_2 \otimes A_3) \otimes (B_2 \otimes B_3)) \arrow[r, "t"] \arrow[d, "1 \otimes (\mu \otimes \mu)"'] & (A_1 \otimes (A_2 \otimes A_3)) \otimes (B_1 \otimes (B_2 \otimes B_3)) \arrow[d, "(1 \otimes \mu) \otimes (1 \otimes \mu)"] & \\
(A_1 \otimes B_1) \otimes (A_{23} \otimes B_{23}) \arrow[r, "t"] & (A_1 \otimes A_{23}) \otimes (B_1 \otimes B_{23}) \arrow[r, "\mu \otimes \mu"] & A_{13} \otimes B_{13}
\end{tikzcd}
$$

Using the definitions of tensor product on categories and $F \otimes G$ on objects, we find that we require an arrow with domain $\mathcal{A}(X_1, X_2) \otimes \mathcal{B}(Y_1, Y_2)$ and codomain $\mathcal{A}'(FX_1, FX_2) \otimes \mathcal{B}'(GY_1, GY_2)$. An obvious choice for such an arrow is $F_{X_1,X_2} \otimes G_{Y_1,Y_2}$, and this is the definition we take. To show that $\otimes$ is functorial is just a matter of expanding definitions.

Furthermore, natural isomorphisms for associativity, left and right identity and symmetry can be defined, thus making $\mathcal{D}!$ into a symmetric monoidal category. I give only the definitions of the morphisms here; the proofs that the components are well-defined $\mathcal{D}$-functors, that they are invertible, and that they have the required naturality and coherence properties are both tedious and unenlightening. The interested reader will surely be able to perform the necessary diagram manipulations without prompting.

Associativity: Let $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ be $\mathcal{D}$-categories. I must define a $\mathcal{D}$-functor

$$\alpha_{\mathcal{ABC}}: (\mathcal{A} \otimes \mathcal{B}) \otimes \mathcal{C} \to \mathcal{A} \otimes (\mathcal{B} \otimes C).$$

Define α to take an object $\langle\langle X, Y\rangle, Z\rangle$ to $\langle X, \langle Y, Z\rangle\rangle$. So each homobject arrow of α must be an arrow (in $\mathcal{D}$) from $(\mathcal{A}(X, X') \otimes \mathcal{B}(Y, Y')) \otimes \mathcal{C}(Z, Z')$ to $\mathcal{A}(X, X') \otimes (\mathcal{B}(Y, Y') \otimes \mathcal{C}(Z, Z'))$. Define this arrow to be the evident component of the associativity transformation of $\mathcal{D}$.

Tensor Identity: Define the tensor identity $\mathcal{I}$ to have a single object (denoted by $*$) and further define $\mathcal{I}(*, *) = I$. $\mu: \mathcal{I}(*, *) \otimes \mathcal{I}(*, *) \to \mathcal{I}(*, *)$ is $\lambda: I \otimes I \to I$ and $i: I \to \mathcal{I}(*, *)$ is the identity arrow at I.

Identity Isomorphisms: The object map of $\lambda: \mathcal{I} \otimes \mathcal{A} \to \mathcal{A}$ sends each object $\langle *, X\rangle$ to X. Thus the homobject arrows must be of the form $\mathcal{I}(*, *) \otimes \mathcal{A}(X, X') \to \mathcal{A}(X, X')$, which is $I \otimes \mathcal{A}(X, X') \to \mathcal{A}(X, X')$. Take this arrow to be $\lambda_{\mathcal{A}(X,X')}$ as defined in $\mathcal{D}$. Similarly, $\rho: \mathcal{A} \otimes \mathcal{I} \to \mathcal{A}$ is defined to send each object $\langle X, *\rangle$ to X and its homobject maps are components $\rho_{\mathcal{A}(X,X')}$ as defined in $\mathcal{D}$.

Symmetry Isomorphism: The object map of $\gamma: \mathcal{A} \otimes \mathcal{B} \to \mathcal{B} \otimes \mathcal{A}$ takes $\langle X, Y\rangle$ to $\langle Y, X\rangle$, and the homobject arrows are $\gamma: \mathcal{A}(X, X') \otimes \mathcal{B}(Y, Y') \to \mathcal{B}(Y, Y') \otimes \mathcal{A}(X, X')$.

This completes the definition of $\mathcal{A} \otimes \mathcal{B}$. The proof that these data define a $\mathcal{D}$-category is omitted.

2.5.1 Examples of tensor product

In the case that $\mathcal{D}$ is cartesian monoidal the tensor product defined above is again categorical product. (This follows from the construction of limits in the proof of theorem 4.12 below.) Thus for sets measured in $\mathbf{2}$, tensor product is the same as product.

More interesting is the tensor product for sets measured in a non-cartesian space, such as $\mathbf{R}_+$ or $\mathbf{R}_+^{\mathrm{op}}$. For both $\mathbf{R}_+!$ and $\mathbf{R}_+^{\mathrm{op}}!$ this tensor product is precisely the orthocurrence operation defined in section 1.2

2.6 MC-categories

In the preceding section it was observed that tensor product of $\mathcal{D}!$ is orthocurrence of sets measured in $\mathcal{D}$. This is now taken further, by defining the other normal operations via purely category-theoretic constructs within $\mathcal{D}!$. Thus, the normal operations can formally be defined on *any* monoidal category possessing certain properties, whether that category arises as $\mathcal{D}!$ for some $\mathcal{D}$ or not. This is very useful because categories of measured sets need not be $\mathcal{D}!$ for any $\mathcal{D}$. In this section I define the normal operations formally, along with a class of categories (the MC categories) to in which those operations are well-defined. In the next section I show that the operations defined here do match their previous definitions (with some minor changes).

In outline: concurrence is coproduct, pairing is product, and exponentiation is internal-hom. Defining concatenation in categorical terms is more difficult, and requires reference to the property of *distributivity*, which will now be defined.

Let i_1 and i_2 denote the left and right injections into a coproduct. In any monoidal category with binary coproducts there exists a unique morphism $\delta\colon (A \otimes C) + (B \otimes C) \to (A + B) \otimes C$ making the following diagram commute.

$$\begin{tikzcd}
A \otimes C \arrow[d, "i_1"'] \arrow[dr, "i_1 \otimes 1"] & \\
(A \otimes C) + (B \otimes C) \arrow[r, dotted, "\delta"] & (A + B) \otimes C \\
B \otimes C \arrow[u, "i_2"] \arrow[ur, "i_2 \otimes 1"'] &
\end{tikzcd}$$

Definition 2.10 A monoidal category with binary coproducts is *distributive* if each such δ is an isomorphism. ∎

Two facts about distributive categories are important. They are summarized in the following two propositions.

Proposition 2.11 *δ is natural in A, B and C.*

Proof: For arrows $f: A \to A'$, $g: B \to B'$ and $h: C \to C'$ I must show commutativity of this diagram:

$$\begin{tikzcd}
(A \otimes C) + (B \otimes C) \arrow[r, "\delta"] \arrow[d, "(f \otimes h) + (g \otimes h)"'] & (A + B) \otimes C \arrow[d, "(f + g) \otimes h"] \\
(A' \otimes C') + (B' \otimes C') \arrow[r, "\delta"] & (A' + B') \otimes C'
\end{tikzcd}$$

By the universal property of coproduct it is sufficient to show that the two routes around the square are equal when composed with the two injections into $(A \otimes C) + (B \otimes C)$. But

$$\begin{array}{rcll}
\delta((f \otimes h) + (g \otimes h))i_1 & = & \delta i_1 (f \otimes h) & i_1 \text{ is natural} \\
& = & (i_1 \otimes 1)(f \otimes h) & \text{definition of } \delta \\
& = & (i_1 f) \otimes h & \otimes \text{ is a functor;}
\end{array}$$

and

$$\begin{array}{rcll}
((f + g) \otimes h)\delta i_1 & = & ((f + g) \otimes h)(i_1 \otimes 1) & \text{definition of } \delta \\
& = & ((f + g) i_1) \otimes h & \otimes \text{ is a functor} \\
& = & (i_1 f) \otimes h & i_1 \text{ is natural.}
\end{array}$$

A similar argument applies for the composition with i_2. ∎

Proposition 2.12 *A closed monoidal category with binary coproducts is distributive.*

I first introduce notation that will shorten this proof and also be helpful in the following chapter when dealing with adjunctions.

Suppose that there is an adjunction $F \dashv R$. For any arrow $g: FA \to B$ denote the adjoint transpose by $g^\sharp: A \to RB$. Dually, denote the adjoint transpose of $f: A \to RB$ by $f^\flat: FA \to B$.

Observe that for appropriate arrows f, g and g' the following equations hold:

$$\begin{array}{rcl}
(g'g)^\sharp & = & (Rg')g^\sharp \\
(g(Ff))^\sharp & = & g^\sharp f.
\end{array}$$

The dual results, $(ff')^\flat = (f^\flat)(Ff')$ and $((Rg')f)^\flat = g'f^\flat$ will also be useful.

Also, given arrows $f: A \to C$ and $g: B \to C$, denote by $(f, g): A + B \to C$ the unique arrow factoring f and g through $A + B$. Thus $(f, g)i_1 = f$ and $(f, g)i_2 = g$ and furthermore any arrow $h: A + B \to C$ can be expressed as (hi_1, hi_2).

Proof: Let $i_1: A \to A+B$ and $i_2: B \to A+B$ be the injections, so that in the notation just introduced, $\delta = (i_1 \otimes 1, i_2 \otimes 1)$. I wish to define $\delta^{-1}: (A+B)\otimes C \to (A\otimes C)+(B\otimes C)$. This is equivalent, using the adjunction between tensor product and exponentiation, to defining an arrow $A+B \to [C, ((A\otimes C)+(B\otimes C))]$. This is in turn equivalent to defining two arrows: one $A \to [C, ((A\otimes C)+(B\otimes C))]$, the other $B \to [C, ((A\otimes C)+(B\otimes C))]$. Using the adjunction in the opposite direction, this is equivalent to defining an arrow $A\otimes C \to (A\otimes C)+(B\otimes C)$ and a corresponding arrow for B. An obvious choice for the last pair of arrows is the injections into the coproduct, which I will denote j_1 and j_2, so that δ^{-1} is defined to be

$$(j_1^\sharp, j_2^\sharp)^\flat: (A + B) \otimes C \to (A \otimes C) + (B \otimes C).$$

I must show that this arrow is in fact $\delta^{-1} = (i_1 \otimes 1, i_2 \otimes 1)^{-1}$.

First observe that

$$(i_1 \otimes 1, i_2 \otimes 1)(j_1^\sharp, j_2^\sharp)^\flat = ([1_C, (i_1 \otimes 1, i_2 \otimes 1)](j_1^\sharp, j_2^\sharp))^\flat.$$

Consider the arrow $[1_C, (i_1 \otimes 1, i_2 \otimes 1)](j_1^\sharp, j_2^\sharp)$ occurring in the above expression. Its domain is $A + B$ and composing with i_1 yields

$$\begin{aligned}
[1_C, (i_1 \otimes 1, i_2 \otimes 1)](j_1^\sharp, j_2^\sharp)i_1 &= [1_C, (i_1 \otimes 1, i_2 \otimes 1)]j_1^\sharp \\
&= ((i_1 \otimes 1, i_2 \otimes 1)j_1)^\sharp \\
&= (i_1 \otimes 1)^\sharp \\
&= (1(i_1 \otimes 1))^\sharp \\
&= 1^\sharp i_1.
\end{aligned}$$

A similar argument shows that

$$[1_C, (i_1 \otimes 1, i_2 \otimes 1)](j_1^\sharp, j_2^\sharp)i_2 = 1^\sharp i_2,$$

hence that

$$[1_C, (i_1 \otimes 1, i_2 \otimes 1)](j_1^\sharp, j_2^\sharp) = 1^\sharp.$$

But then

$$(i_1 \otimes 1, i_2 \otimes 1)(j_1^\sharp, j_2^\sharp)^\flat = 1^{\sharp\flat} = 1,$$

as required.

To show that composition in the other order also yields the identity it is sufficient to show that

$$(j_1^\sharp, j_2^\sharp)^\flat (i_1 \otimes 1, i_2 \otimes 1) j_1 = j_1$$

and

$$(j_1^\sharp, j_2^\sharp)^\flat (i_1 \otimes 1, i_2 \otimes 1) j_2 = j_2.$$

But

$$\begin{aligned} (j_1^\sharp, j_2^\sharp)^\flat (i_1 \otimes 1, i_2 \otimes 1) j_1 &= (j_1^\sharp, j_2^\sharp)^\flat i_1 \otimes 1 \\ &= ((j_1^\sharp, j_2^\sharp) i_1)^\flat \\ &= j_1^{\sharp\flat} \\ &= j_1. \end{aligned}$$

A similar argument for composition with j_2 completes the proof. ∎

I am now able to give two basic definitions.

Definition 2.13 An *MC category* is one that is symmetric monoidal, bicomplete[1] and distributive. An *MCE category* (measured class with exponentiation) is monoidal closed in addition to being MC. ∎

It is important to note that, while categories whose objects are measured classes are important examples of MC categories, they are not the only ones. The structure of objects is important only insofar as it defines a tensor product; the normal operations will be defined below in terms of the tensor product without reference to any object level structure. For example, the cartesian closed ordinals, such as **2**, discussed in the previous section, are MCE by definition though not categories of measured classes.

Define the normal operations on objects of any MC-category as follows.

Definition 2.14

i) *Concurrence* is coproduct;

ii) *Orthocurrence* is tensor product;

iii) *Pairing* is categorical product;

iv) *Constants*: **0** is the initial object, **1** the terminal object and I the tensor identity.

[1] Bicomplete means both complete and cocomplete.

v) *Concatenation.* Concatenation is defined relative to a given object D of the MC category and arrow $d: (I + I) \to D$. The concatenation is defined by this pushout diagram:

$$\begin{tikzcd}
(I+I) \otimes (A \times B) \arrow[r, "e"] \arrow[d, "d \otimes 1"'] & A + B \arrow[d, "i'"] \\
D \otimes (A \times B) \arrow[r, "e'"] & A;^x B
\end{tikzcd}$$

where the map e is the composite

$$(I + I) \otimes (A \times B) \cong (A \times B) + (A \times B) \xrightarrow{\pi_1 + \pi_2} A + B$$

vi) *Local Pairing.* The local operations are defined relative to an *object of locations*, denoted by L and thought of as being the set of locations. A *located object* of an MC category is an object A and an arrow $f: A \to L$. Think of this as an ordinary schedule (A) with an assignment of each event to one of the locations defined by L. I will frequently treat the location map f as understood and refer only to the located object A.

The local pairing of located objects $f: A \to L$ and $g: B \to L$ is their pullback:

$$\begin{tikzcd}
A \times_L B \arrow[r] \arrow[d] & A \arrow[d, "f"] \\
B \arrow[r, "g"] & L
\end{tikzcd}$$

Note that $A \times_L B$ is located in L. Also the concurrence of two located objects can naturally be considered as an object located in L by taking the location map to be the following composite in which Δ denotes the co-diagonal map:

$$A + B \xrightarrow{f+g} L + L \xrightarrow{\Delta} L.$$

vii) *Local Concatenation* is similar to ordinary concatenation but the pairing $A \times B$ appearing in the defining pushout is replaced by a local pairing $A \times_L B$. While this suffices to define the *events* of $A\underline{;}^{x}B$ it does not provide a location map for it. We need an additional map $l\colon D \otimes L \to L$, for then we can define the location map as the composition of l with the map defined in the following diagram.

$$\begin{tikzcd}
(I+I)\otimes(A\times_L B) \arrow[r, "e"] \arrow[d, "d\otimes 1"'] & (A+B) \arrow[r, "f+g"] \arrow[d] & L+L \arrow[d, leftrightarrow] \\
D\otimes(A\times_L B) \arrow[r] \arrow[drr, "1\otimes f\pi_1"'] & A\underline{;}^{x}B \arrow[dr, dotted] & (I+I)\otimes L \arrow[d, "1\otimes d"] \\
 & & D\otimes L
\end{tikzcd}$$

To show that the diagram defines a map I must show that the outer edge commutes. The isomorphism $L + L \to (I + I) \otimes L$ is the composite

$$L + L \xrightarrow{\lambda^{-1}+\lambda^{-1}} (I \otimes L) + (I \otimes L) \xrightarrow{\delta} (I + I) \otimes L.$$

Recall that

$$e = (\pi_1 + \pi_2)(\lambda + \lambda)\delta^{-1}.$$

Hence the composite map around the top and right side of the diagram is

$$(d \otimes 1)\delta(\lambda^{-1} f\pi_1\lambda + \lambda^{-1} g\pi_2\lambda)\delta^{-1}.$$

Naturality of λ means that this is equal to

$$(d \otimes 1)\delta((1 \otimes (f\pi_1)) + (1 \otimes g\pi_2))\delta^{-1}.$$

But $f\pi_1 = g\pi_2$ by definition of local concatenation, so the naturality of δ means this is equal to

$$(d \otimes 1)((1 + 1) \otimes (f\pi_1)\delta\delta^{-1} = d \otimes (f\pi_1),$$

as required.

In most applications of this concept there is no reason for L to enforce constraints on the events of different locations. For example, in the case of $\mathbf{R}_+$-categories L would usually be a set in which all distances were ∞. In such cases $l: D \otimes L \to L$ can simply be defined by $l(d, a) = a$.

viii) *Exponentiation* is defined only if the category in question is MCE, in which case it is defined by the monoidal exponentiation.

ix) The *fusion* defined by maps $f: P \to Q$ and $g: P \to R$ is the pushout of f and g.

∎

2.7 Pomsets as an MC-category

How do the definitions of the preceding section correspond with the definitions of the pomset model, as discussed briefly in section 1.2 and in more detail by Pratt [42]?

The first step in reconciling the set-theoretic notion of pomset with the category-theoretic definitions of the operations must be to define a suitable category of pomsets. Recall that, formally, a pomset is a triple $\langle P, f, \Sigma \rangle$, where P is a partially ordered set and $f: UP \to \Sigma$ is a function from the underlying set of P to Σ.

Definition 2.15 Let **Set** be the category of sets, and **Pos** be the category of posets and monotone maps. Let U: **Pos** $\to$ **Set** be the forgetful functor. The comma category $U \downarrow$ **Set** is the category of pomsets, denoted **Pom**. ∎

To justify this definition, note that the objects of $U \downarrow$ **Set** are triples $\langle P, f, S \rangle$, where P is a poset, S is a set and f:$UP \to S$ is a function. This is the same as the usual definition of pomset as given two paragraphs above. An arrow of this category $\langle P, f, S \rangle \to \langle P', f', S' \rangle$ consists of a monotone map $p: P \to P'$ and a function $s: S \to S'$ such that $sf = f'(Up)$. Considering pomsets as schedules, such a pair maps each event of its domain into an event of the codomain, and each action of the the domain to an action of the codomain. The commutativity condition ensures that if event x is an occurrence of action a then the image of x is an occurrence of the image of a.

The following proposition can be proven from first principles. However I defer proof until section 3.3 where the proof is simplified by use of a general theorem.

Proposition 2.16 **Pom** *is bicomplete. Hence it is an MC category via its cartesian monoidal structure.*

Observe that product and coproduct in **Pom** can be calculated separately in the partial order and the alphabet, so that the coproduct of $\mu: UA \to \Sigma$ and $\mu': UA' \to \Sigma'$ is $\mu + \mu': UA + UA' \to \Sigma + \Sigma'$.

Firstly, it is clear that orthocurrence and pairing, defined category-theoretically, correspond exactly to Pratt's original definition of orthocurrence.

However concurrence, as defined for a general MC category, is not the same as the original pomset definition. In the original definition the alphabet of the result is the *union* of the alphabets of the operands. Under the new definition it is the *disjoint union.* However this new operation, which I will call *disjoint concurrence*, has a reasonable interpretation when measured classes are seen as schedules, since it represents two processes running in parallel without communicating but where each action is now labeled with the identity of the process to which it belongs.

In concatenation the distinction between *disjoint concatenation* and ordinary or union concatenation must also be observed.

First consider the concatenation of posets and later extend to concatenation of pomsets.

Proposition 2.17 *Let A and B be posets and let $d: 1+1 \to 2$ be a monic from the discrete poset with two elements to the ordinal 2. Then $A;^x B$ is the usual concatenation of A and B.*

Proof: The map $e: (1+1) \times A \times B \to A + B$ in the definition of concatenation has the property that

$$\begin{aligned} e(0, a, b) &= a \\ e(1, a, b) &= b \end{aligned}$$

$d \times 1: (1+1) \times A \times B \to 2 \times A \times B$ is easily seen to be the identity map on points.

I must show that the following diagram is a pushout, where the map e' is defined by the same equations as e, and i is the standard inclusion of $A+B$ into $A; B$. Both maps are easily seen to be monotone.

$$\begin{tikzcd} (1+1) \times A \times B \arrow[r, "e"] \arrow[d, "d \times 1"'] & A + B \arrow[d, "i"] \\ 2 \times A \times B \arrow[r, "e'"] & A; B \end{tikzcd}$$

Commutativity is proven by a diagram chase. Next, suppose that there exists a poset X and monotone maps f and g satisfying the appropriate commutativity conditions, and construct a new map j:

$$\begin{tikzcd}
(1+1)\times A\times B \arrow[r, "e"] \arrow[d, "d\times 1"'] & A+B \arrow[d, "i"'] \arrow[ddr, "g"] & \\
2\times A\times B \arrow[r, "e'"] \arrow[drr, "f"'] & A;B \arrow[dr, dotted, "j"'] & \\
& & X
\end{tikzcd}$$

Since i is both 1-1 and onto there exists a unique function j such that $ji = g$. This shows that j is unique. To complete the proof I show that j is monotone and $je' = f$. The fact that $je' = f$ follows immediately the fact that $f(1 \times d) = ge$.

To show monotonicity suppose $x, y \in A; B$ and that $x < y$. If both x and y are in A then $x < y$ is in $A + B$, so $j(x) = g(x) < g(y) = j(y)$. Similar remarks hold if x and y are both in B. The remaining case is $x \in A$ and $y \in B$. But consider the triples $\langle 0, x, y\rangle$ and $\langle 1, x, y\rangle$, both of which are elements of $(1 + 1) \times A \times B$. By definition of e and j it follows that $j(x) = g(x) = ge(\langle 0, x, y\rangle)$ and $j(y) = g(y) = ge(\langle 1, x, y\rangle)$. By the assumption that the outer square of the diagram commutes, we know that $ge = f(d\times 1)$. But $d\times 1$ is the identity function on the underlying sets, hence $j(x) = f(\langle 0, x, y\rangle)$ and $j(y) = f(\langle 1, x, y\rangle)$, where the triples are now considered elements of $2\times A\times B$. But f is monotone and $\langle 0, x, y\rangle < \langle 1, x, y\rangle$, hence $j(x) < j(y)$, as required. This completes the proof. ∎

To extend this result to pomsets first replace the posets 2 and $1 + 1$ by corresponding pomsets, which will be called $\mathbf{2}$ and $|\mathbf{2}|$. Define $|\mathbf{2}|$ to have vertex set $|2|$, alphabet $|2|$ and the identity as labelling function. Define $\mathbf{2}$ to have vertex set 2 and alphabet $|2|$, with labelling function defined by $0 \mapsto 0$ and $1 \mapsto 1$.

It is well known that **Pos** is cartesian closed and bicomplete (so it is an MCE category). However this is not true for **Pom**, as will now be shown.

Proposition 2.18 *Product in* **Pom** *does not preserve coequalizers. Thus* **Pom** *is not cartesian closed.*

Proof: Let E be the empty pomset with singleton alphabet. For any pomset P, $P \times E$ is an empty pomset with an alphabet isomorphic to P. Now consider the linear pomset $a;b;a$ with alphabet $\{a,b\}$, and the single-element pomset a with alphabet $\{a\}$. There are two maps from a to $a;b;a$ that have the inclusion $\{a\} \hookrightarrow \{a,b\}$ as the alphabet map. The coequalizer of these two maps is the singleton pomset with singleton alphabet. (The two occurrences of a are identified and this forces the identification of a with b in the alphabet.) However taking the product of the coequalizer diagram with E yields a diagram of empty pomsets, and the coequalizer of this diagram is the empty pomset with a 2-element alphabet. Thus product with E does not preserve coequalizers. ∎

It will be shown below that replacing **Pos** in the definition of **Pom** by **Pre**, the category of preordered sets, does yield an MCE category.

As noted above the definition of concurrence as coproduct yields disjoint concurrence in **Pom** rather than union concurrence. However, union concurrence can be recovered by considering certain full subcategories of **Pom**.

Definition 2.19

i) $\mathbf{Pom}_\Sigma$ (the category of pomsets with alphabet Σ) is the comma category $V \downarrow \Sigma$. (In this context Σ is the constant functor from the unit category $\mathbf{1}$ to **Set** whose unique value is Σ.

ii) Let **Seti** be the category of sets and inclusion maps, and denote by W the forgetful functor from **Seti** to **Set**. Then the comma category $V \downarrow W$ is denoted **Pomi**.

∎

The proof of the following is once again straightforward, though tedious, from first principles but will follow from results in a later chapter.

Proposition 2.20 *Union concurrence is coproduct in* **Pomi**. *Union concurrence of two pomsets with common alphabet* Σ *is coproduct in* $\mathbf{Pom}_\Sigma$.

Note that, although it would certainly be possible to define a concatenation for **Pomi** and $\mathbf{Pom}_\Sigma$, neither category is monoidal so the general definition cannot be used.

Chapter 3

Construction of MC Categories — Labels

In chapter 2 normal operations were defined for any category of measured classes, provided only that it had the MC or MCE properties. Showing that an arbitrary category is MC or MCE may be quite tedious, but this difficulty need not be too serious because many interesting MC categories can be defined by repeated usage of just two constructions. The first of these constructions takes a category $\mathcal{D}$, considered as a class of measurements, and creates the category $\mathcal{D}!$, which is the category of sets measured in $\mathcal{D}$. An example already seen is the construction of $\mathbf{R}_+$-categories from the category $\mathbf{R}_+$. The second construction begins with a monoidal category $\mathcal{D}$ (one may think of it as a category of measured sets) and a monoidal category $\mathcal{E}$[1], each object of which is considered an alphabet of actions. Its result is $\mathcal{D} \triangleright \mathcal{E}$, each of whose objects is a measured set from $\mathcal{D}$, with elements labeled in an alphabet chosen from $\mathcal{E}$. An example is the construction of **Pom** by taking the measured classes from **Pos** and defining labeling alphabets from **Set**.

This chapter and the next are investigations of the $\triangleright$ and ! operations in turn, aiming to find simple conditions under which the categories constructed are, firstly, bicomplete and closed (and hence MC or MCE) and, secondly, themselves suitable arguments for ! and $\triangleright$.

Examples of the categories which can be defined using this language beginning from the simple monoidal partial orders $\mathbf{1}$, $\mathbf{2}$, $\mathbf{3}$, $\mathbf{3}'$ and $\mathbf{R}_+$ include:

[1]Actually $\mathcal{D}$ and $\mathcal{E}$ must be more than simply monoidal categories. See below.

$1!$	**Set**
$1!!$	**Cat**
$1!!!$	2-categories
$2!$	**Pre**
$2!!$	order-enriched categories
$1! \rhd 1!$	multisets
$2! \rhd 1!$	preordered multisets, (**Prom**)
$\mathbf{3}'!$	prosset orders
$\mathbf{3}'! \rhd 1!$	prossets
$\mathbf{3}! \rhd 1!$	causal/temporal specifications
$\mathbf{R}_+! \rhd 1!$	schedules with minimum delay specifications

Recall from the previous chapter that **Pom** was defined as a comma category. This seems a workable treatment of labeling, but to apply it in the general case both $\mathcal{D}$ and $\mathcal{E}$ must be equipped with suitable functors into a third category. (In the case of **Pom** the third category was **Set**, $\mathcal{D}$ was **Pos** with the forgetful functor to **Set** and $\mathcal{E}$ was **Set** with its identity functor.)

"Suitable" functors turn out to be monoidal functors, and we now recall this definition.

3.1 Monoidal functors

A monoidal functor is intended to be a map between monoidal categories that respects both the categorical and the monoidal structure. It would be natural to define such a functor to be one which preserves tensor product and tensor identity. However, this is found to exclude many functors which are of interest and which enjoy many of the same properties.

It is therefore usual to use a broader definition:

Definition 3.1 Let $\mathcal{D} = \langle \mathcal{D}_0, \otimes, \alpha, \lambda, \rho \rangle$ and $\mathcal{D}' = \langle \mathcal{D}'_0, \otimes', \alpha', \lambda', \rho' \rangle$ be monoidal categories. A monoidal functor from $\mathcal{D}$ to $\mathcal{D}'$ is a triple $\langle F, \phi, \widehat{\phi} \rangle$, where

i) $F \colon \mathcal{D}_0 \to \mathcal{D}'_0$ is a functor;

ii) $\phi_{XY} \colon FX \otimes' FY \to F(X \otimes Y)$ is natural in both variables; and

iii) $\widehat{\phi} \colon I' \to FI$ is an arrow of $\mathcal{D}'$.

F, ϕ and $\widehat{\phi}$ are required to satisfy the following commutativity conditions:

$$\begin{tikzcd}
(FX \otimes' FY) \otimes' FZ \arrow[r, "\phi \otimes' 1"] \arrow[d, "\alpha'"'] & F(X \otimes Y) \otimes' FZ \arrow[r, "\phi"] & F((X \otimes Y) \otimes Z) \arrow[d, "F\alpha"] \\
FX \otimes' (FY \otimes' FZ) \arrow[r, "1 \otimes' \phi"] & FX \otimes' F(Y \otimes Z) \arrow[r, "\phi"] & F(X \otimes (Y \otimes Z))
\end{tikzcd}$$

$$\begin{tikzcd}
FX \otimes' I' \arrow[r, "\rho'"] \arrow[d, "1 \otimes' \widehat{\phi}"'] & FX \\
FX \otimes' FI \arrow[r, "\phi"] & F(X \otimes I) \arrow[u, "F\rho"']
\end{tikzcd}
\qquad
\begin{tikzcd}
I' \otimes' FX \arrow[r, "\lambda'"] \arrow[d, "\widehat{\phi} \otimes' 1"'] & FX \\
FI \otimes' FX \arrow[r, "\phi"] & F(I \otimes X) \arrow[u, "F\lambda"']
\end{tikzcd}$$

If $\mathcal{D}$ and $\mathcal{D}'$ are both symmetric monoidal and the following diagram commutes, we say that $\langle F, \phi, \widehat{\phi} \rangle$ is a symmetric monoidal functor.

$$\begin{tikzcd}
FX \otimes FY \arrow[r, "\phi"] \arrow[d, "\gamma'"'] & F(X \otimes Y) \arrow[d, "F\gamma"] \\
FY \otimes' FX \arrow[r, "\phi"] & F(Y \otimes X)
\end{tikzcd}$$

If ϕ and $\widehat{\phi}$ are isomorphisms we say that $\langle F, \phi, \widehat{\phi} \rangle$ is *strong monoidal.*

If $\langle F, 1, 1 \rangle$ is monoidal we say that F is *strict* monoidal. ∎

There is usually no need to distinguish the tensor products of the domain and codomain of a monoidal functor, so $\otimes'$ is usually denoted by $\otimes$, λ' by λ, and so on.

3.2 Semiconcrete monoidal categories

A category together with a specified functor from it to **Set** is sometimes called a *semiconcrete* category. The functor to **Set** is called the *forgetful* functor. Recall from section 2.2 that **Set** has a cartesian monoidal structure, in which tensor product is categorical product,

I is a singleton, and the associativity and identity morphisms are defined using the universal property of product. So it is possible to talk of monoidal functors to **Set**. A monoidal category will be called semiconcrete if it is equipped with a symmetric monoidal functor to **Set**. In the present context the most important examples of semiconcrete monoidal categories are those where the monoidal category is a category of measured sets and the forgetful functor simply "forgets" the monoidal structure of the set.

More formally, we are concerned with several large categories:

i) The category **SymMon** of small symmetric mononoidal categories and symmetric monoidal functors;

ii) Its subcategory **SSymMon** with the same objects but only the strong symmetric monoidal functors; and

iii) The comma category **SSymMon** $\downarrow$ **Set**. Objects of this category are symmetric monoidal categories together with a strong monoidal functor to **Set**. An arrow from $\langle \mathcal{D}, U_{\mathcal{D}} \rangle$ to $\langle \mathcal{E}, U_{\mathcal{E}} \rangle$ is a strong monoidal functor $F: \mathcal{D} \to \mathcal{E}$ such that $FU_{\mathcal{E}} = U_{\mathcal{D}}$. This category will be denoted by **SSM** (the letters are intended to suggest *s*emiconcrete *s*trong and *m*onoidal).

The choice here of **Set** as the codomain of all forgetful functors is somewhat arbitrary. While the motivating examples can all be expressed in such a framework, it is quite possible that other interesting cases could be obtained by making a different choice. For this reason many of the supporting theorems below are stated and proved in terms of any category.

Definition 3.2 Given a category $\mathcal{D}$ in **SSM** and $\mathcal{E}$ in (**SymMon** $\downarrow$ **Set**), define $\mathcal{D} \rhd \mathcal{E}$ to be the comma category $U_{\mathcal{D}} \downarrow U_{\mathcal{E}}$. Define the forgetful functor of $\mathcal{D} \rhd \mathcal{E}$ to be the composition of $U_{\mathcal{D}}: \mathcal{D} \to$ **Set** with the projection $(U_{\mathcal{D}} \downarrow U_{\mathcal{E}}) \to \mathcal{D}$. ∎

The motivating example for this definition is when $\mathcal{D}$ is a category of measured sets, its forgetful functor forgets the measure structure of a measured set, and $\mathcal{E}$ is **Set** itself. Elements of $\mathcal{D} \rhd \mathcal{E}$ are then composed of a measured set, an alphabet and a function labeling each element of the set with an element of the alphabet.

The remainder of this chapter establishes that $\mathcal{D} \rhd \mathcal{E}$ is in **SSM**, and gives conditions under which it is also an MC category.

3.3 Limits and colimits in comma categories

A discussion of limits or colimits must involve discussion of diagrams. A diagram in a comma category is, by definition, a functor from some index category $\mathcal{I}$ into the comma category. Let us take a closer look at such functors and the natural transformations between them.

For any functor $U: \mathcal{A} \to \mathcal{B}$ denote by $U^{\mathcal{I}}: \mathcal{A}^{\mathcal{I}} \to \mathcal{B}^{\mathcal{I}}$ the functor that maps each $F: \mathcal{I} \to \mathcal{A}$ to $UF: \mathcal{I} \to \mathcal{B}$.

Lemma 3.3 *Let $U: \mathcal{A} \to \mathcal{C}$ and $V: \mathcal{B} \to \mathcal{C}$ be functors. Then*

$$(U \downarrow V)^{\mathcal{I}} \cong (U^{\mathcal{I}} \downarrow V^{\mathcal{I}}).$$

Proof: $U \downarrow V$ is the limit of the following diagram in **Cat** (Mac Lane [31, p. 48]), in which π_1 and π_2 send an arrow to its domain and codomain, respectively.

$$\begin{tikzcd}
 & & \mathcal{B} \arrow[d, "V"] \\
 & \mathcal{C}^{\to} \arrow[r, "\pi_2"] \arrow[d, "\pi_1"] & \mathcal{C} \\
\mathcal{A} \arrow[r, "U"] & \mathcal{C} &
\end{tikzcd}$$

Since the functor $-^{\mathcal{I}}$ is a right adjoint it preserves limits. Note that $(\mathcal{C}^{\to})^{\mathcal{I}}$ is isomorphic to $(\mathcal{C}^{\mathcal{I}})^{\to}$, and that $\pi_1^{\mathcal{I}}: (\mathcal{C}^{\to})^{\mathcal{I}} \to \mathcal{C}^{\to}$ is naturally isomorphic to $\pi_1: (\mathcal{C}^{\mathcal{I}})^{\to} \to \mathcal{C}^{\mathcal{I}}$. Hence $(U \downarrow V)^{\mathcal{I}}$ is the limit of the following diagram.

$$\begin{tikzcd}
 & & \mathcal{B}^{\mathcal{I}} \arrow[d, "V^{\mathcal{I}}"] \\
 & (\mathcal{C}^{\mathcal{I}})^{\to} \arrow[r, "\pi_2"] \arrow[d, "\pi_1"] & \mathcal{C}^{\mathcal{I}} \\
\mathcal{A}^{\mathcal{I}} \arrow[r, "U^{\mathcal{I}}"] & \mathcal{C}^{\mathcal{I}} &
\end{tikzcd}$$

Hence $(U \downarrow V)^{\mathcal{I}}$ is isomorphic to $(U^{\mathcal{I}}) \downarrow (V^{\mathcal{I}})$, as required. ∎

This lemma can be applied whenever functors into a comma category are to be defined. It states that defining a functor into a comma category is essentially equivalent to defining a functor into each of the categories $\mathcal{A}$ and $\mathcal{B}$ and a natural transformation between those functors composed with U and V respectively. Furthermore, natural transformations between such functors are essentially equivalent to a pair of natural transformations satisfying an obvious commutativity condition.

The following theorem is a basic result about limits in comma categories. (Note: I am not aware of this result having appeared in any textbook. The slightly weaker result where $\mathcal{A}$ and $\mathcal{B}$ are assumed complete, V is assumed continuous and $U \downarrow V$ is shown to be complete was proven by Tarlecki in a rather different context [47] and Mac Lane proved a special case as part of his proof of Freyd's Adjoint Functor Theorem [31, page 123].)

Theorem 3.4 *Let $U: \mathcal{A} \to \mathcal{C}$ and $V: \mathcal{B} \to \mathcal{C}$ be functors and $\mathcal{I}$ be a category. If $\mathcal{A}$ and $\mathcal{B}$ have all $\mathcal{I}$-limits and V preserves $\mathcal{I}$-limits then the comma category $(U \downarrow V)$ has all $\mathcal{I}$-limits.*

Proof: Let $F: \mathcal{I} \to (U \downarrow V)$ be an $\mathcal{I}$-diagram in $(U \downarrow V)$. Then by lemma 3.3 F can be considered as a triple $\langle F_1, \mu, F_2 \rangle$ where

$$\begin{aligned} F_1 &: \mathcal{I} \to \mathcal{A} \\ F_2 &: \mathcal{I} \to \mathcal{B} \\ \mu &: UF_1 \to VF_2 \end{aligned}$$

Since $\mathcal{A}$ and $\mathcal{B}$ have $\mathcal{I}$-limits both F_1 and F_2 have limits. Denote their limiting cones by $\lambda^1: \lim F_1 \to F_1$ and $\lambda^2: \lim F_2 \to F_2$ respectively.

Since V preserves $\mathcal{I}$-limits, $V(\lim F_2)$ is a limit of VF_2 with limiting cone $V\lambda^2$.

The composite natural transformation

$$U(\lim F_1) \to U\lambda^1 UF_1 \to \mu VF_2$$

is a cone from $U(\lim F_1)$ to VF_2, so by the universal property of limits there exists a unique arrow p such that the following diagram of functors and natural transformations commutes. (Note: in this and the following diagrams an object of C represents the constant functor to that object and an arrow of C represents the "constant" natural tranformation between such constant functors.)

$$\begin{tikzcd} U(\lim F_1) \arrow[r, "U\lambda^1"] \arrow[d, "p"'] & UF_1 \arrow[d, "\mu"] \\ V(\lim F_2) \arrow[r, "V\lambda^2"] & VF_2 \end{tikzcd}$$

I claim that $\langle \lim F_1, p, \lim F_2 \rangle$ is the limit of F and that the limiting cone is defined by the pair of natural transformations $\langle \lambda^1, \lambda^2 \rangle$.

For suppose $\langle A, f, B\rangle$ ia an object and $\kappa\colon \langle A, f, B\rangle \to F$ is a cone. κ can be considered as a pair of natural transformations $\kappa^1\colon A \to F_1$ and $\kappa^2\colon B \to F_2$. Since F_1 and F_2 have limits there must exist unique maps $a\colon A \to \lim F_1$ and $b\colon B \to \lim F_2$ which factor κ^1 and κ^2 through λ^1 and λ^2 respectively.

Consider the following diagram of natural transformations.

$$\begin{tikzcd} UA \arrow[rrr, "f"] \arrow[dd, "U\kappa^1"'] \arrow[dr, "Ua"] & & & VB \arrow[dd, "V\kappa^2"] \arrow[dl, "Vb"'] \\ & U(\lim F_1) \arrow[r, "p"] \arrow[dl, "U\lambda^1"] & V(\lim F_2) \arrow[dr, "V\lambda^2"'] & \\ UF_1 \arrow[rrr, "\mu"] & & & VF_2 \end{tikzcd}$$

The outer square commutes because κ is an arrow in the comma category. The two inner triangles commute by definition of a and b. The lower inner square commutes by definition of p. I will show that the upper inner square commutes by showing that the cone $(V\lambda^2).(Vb).f$ is identical to the cone $(V\lambda^2).p.(Ua)$. Since there must be a *unique* arrow which factors this cone through the limiting cone $V\lambda^2$, the arrows $(Vb).f$ and $p.(Ua)$ must be identical. This is the required commutativity condition. But then

$$\begin{array}{rcll} (V\lambda^2).(Vb).f & = & (V\kappa^2).f & \text{definition of } b \\ & = & \mu.(U\kappa^1) & \text{outer square commutes} \\ & = & \mu.(U\lambda^1).(Ua) & \text{definition of } a \\ & = & (V\lambda^2).p.(Ua) & \text{lower square commutes} \end{array}$$

This proves the required commutativity, which shows that $\langle a, b\rangle$ is an arrow in $U \downarrow V$ and that it factors κ. That $\langle a, b\rangle$ is the unique such arrow follows from the fact that a

factors κ^1 through the limiting cone λ^1 so a is the unique such arrow, and b factors κ^2 through λ^2, hence b is also unique. ∎

This proof gives an explicit construction of limits in a comma category when the conditions of the theorem are satisfied.

Corollary 3.5 *If $\mathcal{A}$ and $\mathcal{B}$ have all $\mathcal{I}$-colimits and U preserves $\mathcal{I}$-colimits then $U \downarrow V$ has all $\mathcal{I}$-colimits.*

Proof: (Due to Roger Crew.) For any functor $U: \mathcal{A} \to C$ let U^{op} be the corresponding functor $U^{\mathrm{op}}: \mathcal{A}^{\mathrm{op}} \to C^{\mathrm{op}}$. Observe that

$$(U \downarrow V)^{\mathrm{op}} = (V^{\mathrm{op}} \downarrow U^{\mathrm{op}}).$$

∎

An application of these results is to prove proposition 2.16 of section 2.7, which stated that **Pom**, the category of pomsets, is bicomplete.

Proof of proposition 2.16: Recall that $\mathbf{Pom} = U \downarrow \mathbf{Set}$, where $U: \mathbf{Pos} \to \mathbf{Set}$ is the forgetful functor. **Pos** and **Set** are both bicomplete. The identity functor on **Set** preserves limits, hence **Pom** is complete by the theorem. U preserves coproducts, so **Pom** has all coproducts.

It remains to show that **Pom** posesses coequalizers but, since U does not preserve them, the corollary will not help in this regard and the result is obtained by constructing coequalizers in **Pom** explicitly.

(The following construction is due to José Meseguer.)

Suppose that

$$\langle f, i\rangle, \langle g, j\rangle: \langle A, \mu, \Sigma\rangle \to \langle A', \mu', \Sigma'\rangle$$

are two parallel arrows in **Pom**: Let the coequalizer of f and g in **Pos** be $h: A' \to B$ and let the coequalizer of i and j in **Set** be $k: \Sigma' \to T$. Define a set Λ and maps m and ν as a pushout in **Set**:

$$\begin{tikzcd}
UA \arrow[r, "\mu"] \arrow[d, shift right, "Uf"'] \arrow[d, shift left, "Ug"] & \Sigma \arrow[d, shift right, "i"'] \arrow[d, shift left, "j"] & \\
UA' \arrow[r, "\mu'"] \arrow[d, "Uh"'] & \Sigma' \arrow[r, "k"] & T \arrow[d, "m"] \\
UB \arrow[rr, "\nu"] & & \Lambda
\end{tikzcd}$$

To see that $\langle B, \nu, \Lambda\rangle$ is the coequalizer of $\langle f, i\rangle$ and $\langle g, j\rangle$ observe, firstly, that $\langle h, mk\rangle$ does in fact equalize the arrows $\langle f, i\rangle$ and $\langle g, j\rangle$ by construction.

Suppose now that an arrow $\langle h', k'\rangle\colon \langle A', \mu', \Sigma'\rangle \to \langle B', \nu', \Lambda'\rangle$ equalizes $\langle f, i\rangle$ and $\langle g, j\rangle$.

(Refer to the diagram below.) I now construct an arrow from $\langle B, \nu, \Lambda\rangle$ to $\langle B', \nu', \Lambda'\rangle$. To do this note that since h' equalizes f and g, and h is the coequalizer of f and g, there is a unique map p such that

$$ph = h'.$$

Similarly, k' equalizes i and j there is a unique map q such that

$$qk = k'.$$

Now I wish to define an arrow from Λ to Λ'. Recall that Λ was defined as the pushout of Uh and $k\mu'$. If I can show that $\nu'(Up)(Uh) = qk\mu'$ then the universal property of pushouts will give a unique arrow from Λ to Λ'. But

$$\begin{aligned}
\nu'(Up)(Uh) &= \nu'(Uh') && \text{(definition of } h'\text{)}\\
&= k'\mu' && \text{(}\langle h', k'\rangle \text{ is an arrow in } \mathbf{Pom}\text{)}\\
&= qk\mu' && \text{(definition of } k'\text{)}
\end{aligned}$$

Hence there exists an arrow r which makes everything commute. The proof is completed by a routine verification that $\langle p, r\rangle$ is an arrow in **Pom** and is unique.

$$
\begin{tikzcd}
UA \arrow[rrrr, "\mu"] \arrow[d, shift right, "Uf"'] \arrow[d, shift left, "Ug"] & & & & \Sigma \arrow[d, shift right, "i"'] \arrow[d, shift left, "j"] \\
UA' \arrow[rrrr, "\mu'"] \arrow[dr, "Uh"] \arrow[dd, "Uh'"'] & & & & \Sigma' \arrow[dl, "k"'] \arrow[dd, "k'"] \\
& UB \arrow[r, "\nu"] \arrow[dl, "Up"] & \Lambda \arrow[drr, "r"'] & T \arrow[l, "m"'] \arrow[dr, "q"] & \\
UB' \arrow[rrrr, "\nu'"] & & & & \Lambda'
\end{tikzcd}
$$

∎

3.4 Exponentiation in comma categories

In the definition of **Pom**, **Pos** can be replaced by **Pre**, the category of preordered sets, to yield **Prom**, the category of preordered multisets. Although **Pom** and **Prom** appear quite similar there is an important difference: **Prom** is cartesian closed, while **Pom** is not (the negative result was shown in proposition 2.18).

It turns out that closure is assured by the existence of a right adjoint to the forgetful functor from **Pre** to **Set**. (This right adjoint takes any set to the *clique* on that set, that is, the preorder defined by setting $x \leq y$ for every x and y in the set.) This can be formalized in a general way as the next theorem.

Theorem 3.6 *Let $\mathcal{A}$ and $\mathcal{B}$ be monoidal closed categories and let $\mathcal{C}$ be a monoidal category. Suppose further that $F: \mathcal{A} \to \mathcal{C}$ is a strong monoidal functor, that $G: \mathcal{B} \to \mathcal{C}$ is (not necessarily strong) monoidal and that $\mathcal{A}$ has all pullbacks. If F admits a right adjoint R then $F \downarrow G$ is closed.*

The proof can be divided into three lemmas. I will show that

i) R is monoidal and hence that RG is monoidal;

ii) $F \downarrow G$ is isomorphic to $1_{\mathcal{A}} \downarrow RG$; and finally

iii) For any monoidal functor $H: \mathcal{A} \to \mathcal{B}$, $1_{\mathcal{A}} \downarrow H$ is closed.

The proofs make frequent use of the $\sharp$ and $\flat$ notation introduced in the proof of proposition 2.12.

Lemma 3.7 *Let $F: \mathcal{A} \to \mathcal{C}$ be strong monoidal and have a right adjoint R. Then R is monoidal.*

Proof: Let $\phi: FA \otimes FB \to F(A \otimes B)$ and $\widehat{\phi}: I \to FI$ be the isomorphisms which make F monoidal.

Define $\psi: RX \otimes RY \to R(X \otimes Y)$ to be the adjoint transpose of the composite

$$F(RX \otimes RY) \xrightarrow{\phi^{-1}} FRX \otimes FRY \xrightarrow{\epsilon \otimes \epsilon} X \otimes Y;$$

and $\widehat{\psi}: I \to RI$ to be $(\widehat{\phi}^{-1})^{\sharp}$.

To show that ψ and $\widehat{\psi}$ make R into a monoidal functor I must show that they satisfy the coherence conditions. I give here the proof of the associativity condition. The proofs of the other two conditions are similar.

F is monoidal, so we know that the following commutes:

$$\begin{tikzcd}
(FRX \otimes FRY) \otimes FRZ \arrow[r, "\alpha"] \arrow[d, "\phi \otimes 1"'] & FRX \otimes (FRY \otimes FRZ) \arrow[d, "1 \otimes \phi"] \\
F(RX \otimes RY) \otimes FRZ \arrow[d, "\phi"'] & FRX \otimes F(RY \otimes RZ) \arrow[d, "\phi"] \\
F((RX \otimes RY) \otimes RZ) \arrow[r, "F\alpha"] & F(RX \otimes (RY \otimes RZ))
\end{tikzcd}$$

Rearrange to obtain the equation

$$\alpha(\phi^{-1} \otimes 1)\phi^{-1} = (1 \otimes \phi^{-1})\phi^{-1} F\alpha.$$

Compose both sides of the equation with $\epsilon \otimes (\epsilon \otimes \epsilon)$. Then the left hand side of the new equation is:

$$\begin{aligned}
& (\epsilon \otimes (\epsilon \otimes \epsilon))\alpha(\phi^{-1} \otimes 1)\phi^{-1} \\
= \quad & \alpha(\psi^{\flat} \otimes \epsilon)\phi^{-1}
\end{aligned}$$

$$\begin{aligned} &= \alpha((\epsilon(F\psi)) \otimes \epsilon)\phi^{-1} \\ &= \alpha(\epsilon \otimes \epsilon)\phi^{-1}F(\psi \otimes 1) \\ &= \alpha\psi^\flat F(\psi \otimes 1) \end{aligned}$$

The right side of the new equation is

$$\begin{aligned} &(\epsilon \otimes (\epsilon \otimes \epsilon))(1 \otimes \phi^{-1})\phi^{-1}(F\alpha) \\ &\quad = (\epsilon \otimes \psi^\flat)\phi^{-1}(F\alpha) \\ &\quad = (\epsilon \otimes (\epsilon(F\psi)))\phi^{-1}(F\alpha) \\ &\quad = (\epsilon \otimes \epsilon)\phi^{-1}F(1 \otimes \psi)(F\alpha) \\ &\quad = \psi^\flat F(1 \otimes \psi)(F\alpha) \end{aligned}$$

Apply $\sharp$ to both sides of the new equation and simplify using the properties of $\sharp$ to obtain the required condition

$$(R\alpha)\psi(\psi \otimes 1) = \psi(1 \otimes \psi)\alpha.$$

∎

Lemma 3.8 *Let $R: \mathcal{C} \to \mathcal{A}$ be a right adjoint of F, considered as monoidal functor as described in the previous lemma. Under the conditions of the theorem $F \downarrow G$ and $1_{\mathcal{A}} \downarrow RG$ are isomorphic monoidal categories.*

Proof: Map each object $FA \xrightarrow{f} GB$ to the corresponding object $A \xrightarrow{f^\sharp} RGB$. It is straightforward to show that this object map can be extended to an isomorphism of categories.

Now I show that the tensor product of the two categories coincide.

Let γ be the natural transformation $GB_1 \otimes GB_2 \to G(B_1 \otimes B_2)$. Let ϕ be the natural isomorphism $FA_1 \otimes FA_2 \to F(A_1 \otimes A_2)$. Denote the corresponding transformation for R by ψ, recalling that $\psi = ((\epsilon \otimes \epsilon)\phi^{-1})^\sharp$. Hence the transformation for the composite functor RG is $(R\gamma)\psi$.

Consider two objects $FA_1 \xrightarrow{f_1} GB_1$ and $FA_2 \xrightarrow{f_2} GB_2$. Applying $\sharp$ and then taking the tensor product in $1 \downarrow RG$ results in the composite arrow

$$A_1 \otimes A_2 \xrightarrow{f_1^\sharp \otimes f_2^\sharp} RGB_1 \otimes RGB_2 \xrightarrow{(R\gamma)\psi} RG(B_1 \otimes B_2).$$

I want to show that this arrow is identical to the result of first taking tensor product in $F \downarrow G$ and then applying $\sharp$, namely

$$(\gamma(f_1 \otimes f_2)\phi^{-1})^\sharp.$$

But

$$\begin{array}{ll} (R\gamma)\psi(f_1^\sharp \otimes f_2^\sharp) & \\ = (R\gamma)((\epsilon \otimes \epsilon)\phi^{-1})^\sharp(f_1^\sharp \otimes f_2^\sharp) & \text{definition of } \psi \\ = (\gamma(\epsilon \otimes \epsilon)\phi^{-1}F(f_1^\sharp \otimes f_2^\sharp))^\sharp & \text{properties of } \sharp \\ = (\gamma(\epsilon \otimes \epsilon)(F(f_1^\sharp) \otimes F(f_2^\sharp))\phi^{-1})^\sharp & \phi \text{ is natural} \\ = (\gamma(\epsilon F(f_1^\sharp) \otimes \epsilon F(f_2^\sharp))\phi^{-1})^\sharp & \\ = (\gamma(f_1 \otimes f_2)\phi^{-1})^\sharp. & \end{array}$$

Which is the required result.

∎

Lemma 3.9 *Let $\mathcal{A}$ and $\mathcal{B}$ be closed monoidal categories and $H\colon \mathcal{A} \to \mathcal{B}$ be a monoidal functor. Suppose that $\mathcal{A}$ has pullbacks. Then the monoidal category $1_{\mathcal{A}} \downarrow H$ is closed.*

Proof: To each object $FA \xrightarrow{f} GB$ associate the corresponding object $A \xrightarrow{f^\sharp} RGB$. It is straightforward to show that this correspondence can be extended to an isomorphism of categories.

Now I show that the tensor product of the two categories coincide.

Let γ be the natural transformation $GB_1 \otimes GB_2 \to G(B_1 \otimes B_2)$. Let ϕ be the natural isomorphism $FA_1 \otimes FA_2 \to F(A_1 \otimes A_2)$. Denote the corresponding transformation for R by ψ, recalling that $\psi = ((\epsilon \otimes \epsilon)\phi^{-1})^\sharp$. Hence the transformation for the composite functor RG is $(R\gamma)\psi$.

Consider two objects $FA_1 \xrightarrow{f_1} GB_1$ and $FA_2 \xrightarrow{f_2} GB_2$. Applying $\sharp$ and then taking the tensor product in $1 \downarrow RG$ results in the composite arrow

$$A_1 \otimes A_2 \xrightarrow{f_1^\sharp \otimes f_2^\sharp} RGB_1 \otimes RGB_2 \xrightarrow{(R\gamma)\psi} RG(B_1 \otimes B_2).$$

I want to show that this arrow is identical to the result of first taking tensor product in $F \downarrow G$ and then applying $\sharp$, namely

$$(\gamma(f_1 \otimes f_2)\phi^{-1})^\sharp.$$

But

$$
\begin{aligned}
&(R\gamma)\psi(f_1^\sharp \otimes f_2^\sharp) \\
&= (R\gamma)((\epsilon \otimes \epsilon)\phi^{-1})^\sharp(f_1^\sharp \otimes f_2^\sharp) && \text{definition of } \psi \\
&= (\gamma(\epsilon \otimes \epsilon)\phi^{-1}F(f_1^\sharp \otimes f_2^\sharp))^\sharp && \text{properties of } \sharp \\
&= (\gamma(\epsilon \otimes \epsilon)(F(f_1^\sharp) \otimes F(f_2^\sharp))\phi^{-1})^\sharp && \phi \text{ is natural} \\
&= (\gamma(\epsilon F(f_1^\sharp) \otimes \epsilon F(f_2^\sharp))\phi^{-1})^\sharp \\
&= (\gamma(f_1 \otimes f_2)\phi^{-1})^\sharp.
\end{aligned}
$$

Which is the required result.

∎

3.5 A counterexample

For all the categories considered so far any right adjoint to the forgetful functor has the property that it is a one-sided inverse to the forgetful functor. (In other words, **Set** may be considered a coreflective subcategory.) It is natural to ask whether this need always be the case. In this section I present a counterexample.

Definition 3.10 A *graded set* is a pair $\langle A, \mathrm{gr}\rangle$, where A is a set and gr is a function from A to the natural numbers. For each $a \in A$ denote by $\mathrm{gr}(a)$ the *grade of a*. Denote by A_n the set $\{ a \in A \mid \mathrm{gr}(a) = n \}$. $\langle A, \mathrm{gr}\rangle$ will usually be abbreviated to A.

If A and B are graded sets a *morphism of graded sets* between them is a function $f\colon A \to B$ such that for all $a \in A$, $\mathrm{gr}(f(a)) = \mathrm{gr}(a)$. Clearly this makes the class of graded sets into a category, which will be called **GSet**. ∎

First some basic facts about **GSet**.

Proposition 3.11 **GSet** *is complete, cocomplete and cartesian closed. Furthermore,* **GSet** *can be construed as an object of* **SSM** *by taking the forgetful functor U defined by $UA = A_0$.*

Proof: It is easy to see that **GSet** is isomorphic to $\mathbf{Set}^{\mathbf{N}}$, where **N** is the discrete category whose objects are the natural numbers. Any power of **Set** is a topos, hence complete, cocomplete and cartesian closed (see, for example, Goldblatt [23]).

As an alternative, more elementary proof note that limits and colimits can be calculated "pointwise" at each object of **N**. In other words they can be calculated separately for each grade so that, for example,

$$(A \times B)_n = A_n \times B_n.$$

In view of this it is easy to see that exponentials can also be calculated separately for each grade. ∎

Recall that $\mathbf{R}_+^{\mathrm{op}}$ was both cartesian closed and closed with respect to another tensor product. **GSet** has a similar property, as is shown in the next two propositions.

Definition 3.12 The *tensor product* of two graded sets A and B has underlying set $A \times B$ and grading function defined by

$$\mathrm{gr}(\langle a, b\rangle) = \mathrm{gr}(a) + \mathrm{gr}(b).$$

∎

Proposition 3.13 **GSet**, *considered as a monoidal category under the tensor product just defined, is closed.*

Proof: I give a right adjoint for tensor product explicitly. Given graded sets A and B define the graded set $[A, B]$ to consist of all functions f from A to B (*not* only morphisms of graded sets) such that for all $a \in A$

$$\mathrm{gr}(f(a)) = \mathrm{gr}(a) + n.$$

So f "shifts up" the grade of each element of A by a fixed amount, and that fixed amount is the grade of f. Define $[-, -]$ on arrows to be composition in a standard way. This is clearly right adjoint to tensor product. ∎

Proposition 3.14 **GSet** *admits two distinct strict monoidal functors to* **Set**.

Proof: One functor simply forgets the grading and takes the graded set A to the ordinary set A. The other takes the graded set A to A_0. ∎

Finally, here is the proof that **GSet** is the counterexample that was sought.

Proposition 3.15 *The forgetful functor U from* **GSet** *(under tensor product) to* **Set**, *taking each graded set to its underlying set, has a right adjoint R, but $UR \neq 1_{\mathbf{Set}}$.*

Proof: Define RX by setting $(RX)_n = X$ for all grades n. It is easy to show that there exists a set map from a graded set A to X if and only if there exists a graded map from A to RX, so that R is the right adjoint of U. Clearly, however $URX = X \times \mathbf{N}$, which is not equal to X in general. ∎

3.6 The forgetful functor of $\mathcal{D} \rhd \mathcal{E}$

In this section I investigate conditions under which the $\rhd$ operation can be iterated. If $\mathcal{D}$ and $\mathcal{E}$ are bicomplete closed and strong symmetric monoidal then theorem 3.4, corollary 3.5 and theorem 3.6 show that these properties are preserved whenever appropriate adjoints to the forgetful functors $U_{\mathcal{D}}$ and $U_{\mathcal{E}}$ exist. So one approach is to construct adjoints for $U_{\mathcal{D} \rhd \mathcal{E}}$, recalling that this forgetful functor is defined to be the composite functor

$$(\mathcal{D} \rhd \mathcal{E}) \xrightarrow{\pi} \mathcal{D} \xrightarrow{U_{\mathcal{D}}} \mathbf{Set}.$$

Proposition 3.16 *If $\mathcal{E}$ has a terminal object, $U_{\mathcal{D}}$ has a right adjoint and $U_{\mathcal{E}}$ has a left adjoint then $U_{\mathcal{D} \rhd \mathcal{E}}$ also has a right adjoint.*

Proof: Let $R\colon \mathbf{Set} \to \mathcal{D}$ be right adjoint to $U_{\mathcal{D}}$. The desired right adjoint maps each set X to the object $U_{\mathcal{D}}RX \to U_{\mathcal{E}}1$ — note that there is a unique such arrow because $U_{\mathcal{E}}$ preserves limits, in particular it preserves terminal objects. The counit of the adjunction is given by the map

$$U_{\mathcal{D} \rhd \mathcal{E}}(U_{\mathcal{D}}RX \to U_{\mathcal{E}}1) = U_{\mathcal{D}}RX \xrightarrow{\epsilon} X.$$

∎

Thus, an expression of the form

$$(\ldots((\mathcal{D} \rhd \mathcal{E}_1) \rhd \mathcal{E}_2) \rhd \ldots) \rhd \mathcal{E}_n$$

denotes a bicomplete closed strong symmetric monoidal category provided that $\mathcal{D}$ and all the $\mathcal{E}_i$ have these properties, $U_{\mathcal{D}}$ has a right adjoint and each $U_{\mathcal{E}_i}$ has a left adjoint.

Chapter 4

Construction of MC Categories — Enrichment

Enrichment, the operation taking a symmetric monoidal category $\mathcal{D}$ to $\mathcal{D}!$, was introduced in chapter 2. In this chapter I continue the program outlined at the beginning of chapter 3, aimed at deriving properties of $\mathcal{D}!$ from those of $\mathcal{D}$. The first task is to give conditions on $\mathcal{D}$ that will ensure $\mathcal{D}!$ is complete, cocomplete or closed.

I also observe that $\mathcal{D}!$ always has a strong monoidal functor to **Set** so that, in addition to its independent use as an MC category, it may be used as an argument of $\triangleright$. In order to apply the results of section 3.3, it is important to know not just that limits and colimits exist in $\mathcal{D}!$ but whether they are preserved by its forgetful functor. Similarly, by the results of section 3.4 the existence of a right adjoint to the forgetful functor can be used to show closure of a comma category. In this chapter I show that the existence of adjoints (hence the preservation of limits and colimits) can be assured by some relatively simple conditions on $\mathcal{D}$.

Next, I briefly discuss other means of defining MC categories.

Finally, I consider expressions formed by repeated use of $\triangleright$ and !, giving simple conditions under which such expressions are well-formed and denote MCE categories.

4.1 Enrichment as a functor

A preliminary technical step in deriving completeness properties of $\mathcal{D}!$ is to observe that the definition of ! can be extended to symmetric monoidal functors and that if F is a strong monoidal functor so is $F!$. That is, ! can be seen as an endofunctor on **SymMon**, or as an endofunctor on **SSymMon**. Furthermore, **SymMon** has a natural 2-category structure and ! is also 2-functor.

The functoriality of ! also holds some interest independent of this application. A monoidal functor can be seen as a translation between measurement systems, and applying ! to it yields a corresponding translation between sets measured in each system. For example, a monoidal functor from $\mathbf{R}_+$ to $\mathbf{2}$ generates a translation from real-time schedules to partial-order schedules. It often happens that only part of a concurrent system requires attention to timing details, while the remainder can be specified in terms of the partial order of events. The functor translating from real-time to partial orders may find useful application in relating the timing-dependent and timing-independent parts of a system specification.

In outline, given a symmetric monoidal functor $F: \mathcal{D} \to \mathcal{E}$, $F!: \mathcal{D}! \to \mathcal{E}!$ will take each $\mathcal{D}$-category $\mathcal{A}$ to an $\mathcal{E}$-category having the same objects as $\mathcal{A}$ and homobjects given by $((F!)\mathcal{A})(X,Y) = F(\mathcal{A}(X,Y))$. Before discussing the technicalities involved in this definition I fix ideas with some examples of the action of !.

i) $F: \mathbf{3}' \to \mathbf{2}$ defined by $F0 = 0$, $F1 = F2 = 1$ is monoidal. $F!$ takes a prosset order to a preorder by erasing the distinction between weak and strong ordering of events. (The only other monoidal functor from $\mathbf{3}'$ to $\mathbf{2}$ is the constant functor to 1. When lifted to schedules, this functor forgets all ordering information.)

ii) There are two nonconstant monoidal functors from $\mathbf{3}$ to $\mathbf{2}$. One is defined by $F0 = 0$ and $F1 = F2 = 1$, and $F!$ converts a schedule in $\mathbf{3}!$ to a preorder by erasing the distinction between temporal and causal ordering. The other is defined by $F0 = F1 = 0$ and $F2 = 1$, and $F!$ converts a $\mathbf{3}$-schedule to a preorder by ignoring the temporal order completely. This generalizes to any finite ordinal considered as a cartesian closed category. There are precisely n monoidal functors from n to 2 because there are $n+1$ monotone functions, but the requirement that they be monoidal rules out the constant functor to 0. Each such functor corresponds to ignoring all constraints weaker than a certain level.

iii) The function "multiplication by a" is monoidal from $\mathbf{R}_+$ to $\mathbf{R}_+$ for any positive a. Its corresponding functor on schedules maintains the same events but multiplies all time measurements by a, which can be seen either as slowing time by a factor of a or as changing the units in which time is measured. (All monoidal functors from $\mathbf{R}_+$

to $\mathbf{R}_+$ have the form of multiplication by a positive constant. To see this, first note that if 1 is sent to a then any integer n must be sent to an because the functor is monoidal. From this the result extends to all rationals, and then functoriality extends it to all real numbers. This can be stated as $[\mathbf{R}_+, \mathbf{R}_+] \cong \mathbf{R}_+$ in the category of small symmetric monoidal categories.)

iv) $F: \mathbf{R}_+ \to \mathbf{3}'$ defined by $F(-\infty) = 0$, $F0 = 1$ and $FX = 2$ for other values of X induces a mapping from schedules with lower bound constraints to prosset orders. This induces quite a natural map $F!: \mathbf{R}_+! \to \mathbf{3}'!$ between the two categories of schedules, since two events separated by a 0 lower bound may occur simultaneously and so can be considered as weakly ordered. Other constraints do not allow simultaneity and must be considered as strong orderings.

v) $F: \mathbf{R}_+^{\mathrm{op}} \to \mathbf{3}$ defined by $F\infty = 0$, $F0 = 2$ and $FX = 1$ otherwise, induces a map $F!: \mathbf{R}_+^{\mathrm{op}}! \to \mathbf{3}!$ from schedules with upper bound constraints to schedules with temporal and causal orderings. Under this translation event A causes event B only if the distance from A to B is zero. That is the mapping defines "A causes B" to mean "there can be no gap between A and B."

vi) The map from $\mathbf{2}$ to $\mathbf{Set}$ defined by sending 0 to the empty set and 1 to a singleton induces the standard mapping under which a preorder is considered a category with at most a single map in any homset.

The first technical observation to be made is that every $\mathcal{D}!$ has a natural forgetful functor.

Lemma 4.1 *Let $\mathcal{D}$ be a symmetric monoidal category. $\mathcal{D}!$ and has a strong monoidal functor to* **Set***.*

Proof: We have already seen that $\mathcal{D}!$ is symmetric monoidal. Define a functor from $\mathcal{D}!$ to **Set** to take each $\mathcal{D}$-category to its set of objects. By the definition of tensor product in $\mathcal{D}!$ this forgetful functor is strong monoidal. ∎

The definition of $F!: \mathcal{D}! \to \mathcal{E}!$ on the objects of $\mathcal{D}!$ (that is, on $\mathcal{D}$-categories) has been stated earlier in this section. It remains to define $F!$ on the arrows of $\mathcal{D}!$. So suppose that $P: \mathcal{A} \to \mathcal{B}$ is a $\mathcal{D}$-functor. Define $(F!)P$ (an $\mathcal{E}$ functor) to have the same object map as P and to be defined on homobjects by taking $(F!)_{AA'}$ to be the arrow $F(P_{AA'})$.

Recall that **SymMon** denotes the category of (small) symmetric monoidal categories and symmetric monoidal functors and **SSM** the category of semiconcrete monoidal categories and strong monoidal functors. I now show that $F!$ is well-behaved with respect to the forgetful functors of its domain and codomain.

Lemma 4.2 *If* $F: \mathcal{D} \to \mathcal{E}$ *is a morphism of* **SymMon** *then* $F!: \mathcal{D}! \to \mathcal{E}!$ *is a morphism of* **SSM**.

Proof: For the proof that $F!$ is strong symmetric monoidal see Eilenberg and Kelly [17]. It remains to show that $U_{\mathcal{E}!}F! = U_{\mathcal{D}!}$. Note that this is an equation between monoidal functors; proof of equality must include proof that the associated natural transformations are equal.

Let $\mathcal{A}$ be a $\mathcal{D}$-category. Then $U_{\mathcal{E}!}(F!)\mathcal{A}$ is the set of objects of $F!\mathcal{A}$, which is the same as the set of objects of $\mathcal{A}$. This is equal to $U_{\mathcal{D}!}\mathcal{A}$.

Let $P: \mathcal{A} \to \mathcal{B}$ be a $\mathcal{D}$-functor. Then $U_{\mathcal{E}!}F!P$ is the object map of $F!P$, which is the same as the object map of P. This is equal to $U_{\mathcal{D}!}P$. Hence the two functors are equal.

Now $U_{\mathcal{D}!}$ and $U_{\mathcal{E}!}$ are strict monoidal. To show the equality of the required natural transformations I must show that

$$(U_{\mathcal{E}!}\phi^{F!})1 = 1.$$

But the object map of the $\mathcal{E}$-functor $\phi^{F!}$ is the identity map by definition, so the required equality holds there. A similar argument holds for $\widehat{\phi}: \mathcal{I} \to (F!)\mathcal{I}$. ∎

One need only check that $(GF)! = (G!)(F!)$ and $1! = 1$, both easy exercises, to establish:

Theorem 4.3 ! *defines a functor from* **SymMon** *to* **SSM**.

Of course one can view ! as an endofunctor on **SymMon** rather than a functor into **SSM**. (The endofunctor is actually $!$: **SymMon** $\to$ **SSM** composed with the standard projection functor **SSM** $\to$ **SSymMon** then composed with the inclusion **SSymMon** $\to$ **SymMon**.) All these functors will be denoted by !. Which ! is intended will be clear from the context.

The category of monoidal categories and monoidal functors is made into a 2-category by defining 2-cells to be those natural transformations that, informally speaking, respect the monoidal structure of the functors. Formally, define as follows.

Definition 4.4 A *monoidal natural transformation* between monoidal functors $(F, \phi, \widehat{\phi})$ and $(G, \psi, \widehat{\psi})$ is defined to be a natural transformation $\nu: F \to G$ such that the following diagrams commute

$$\begin{tikzcd}
FX \otimes FY \arrow[r, "\phi"] \arrow[d, "\nu \otimes \nu"'] & F(X \otimes Y) \arrow[d, "\nu"] \\
GX \otimes GY \arrow[r, "\psi"] & G(X \otimes Y)
\end{tikzcd}$$

$$\begin{tikzcd}
I \arrow[r, "\widehat{\phi}"] \arrow[dr, "\widehat{\psi}"'] & FI \arrow[d, "\nu"] \\
& GI
\end{tikzcd}$$

∎

The following proposition, ensuring that **SymMon** can in fact be seen as a 2-category, is then easily obtained.

Proposition 4.5 *i) Identity natural transformations are monoidal;*

ii) Horizontal and vertical compositions of monoidal natural transformations yield monoidal natural transformations.

iii) Hence monoidal categories, monoidal functors and monoidal natural transformations form a 2-category.

The categories **SymMon**, **SSymMon** and **SSM** can all be seen as 2-categories that inherit this notion of monoidal natural transformation.

The functor ! now extends to a 2-functor.

Theorem 4.6 ! *extends to a 2-functor on the 2-category of symmetric monoidal categories, functors and natural transformations.*

Proof: Let $(F, \phi, \widehat{\phi}), (G, \psi, \widehat{\psi}) \colon \mathcal{D} \to \mathcal{E}$ be symmetric monoidal functors and $\nu \colon F \to G$ be monoidal natural. Define $\nu! \colon F! \to G!$ by defining the component at each $\mathcal{D}$-category $\mathcal{A}$, that is $\nu!_{\mathcal{A}}\colon (F!)\mathcal{A} \to (G!)\mathcal{A}$, to be the identity on objects and for objects X and Y of $\mathcal{A}$ (hence objects of $(F!)\mathcal{A}$ and $(G!)\mathcal{A}$)

$$(\nu!_{\mathcal{A}})_{XY} = \nu_{\mathcal{A}(X,Y)}.$$

The proof is completed by checking a number of conditions. $\nu!_{\mathcal{A}}$ is an $\mathcal{E}$-functor because ν is monoidal and natural. $\nu!$ is natural in $\mathcal{A}$ owing to naturality of ν. $\nu!$ being monoidal follows from monoidality of ν and the definitions of $\phi!$ and $\psi!$. That ! preserves identities, horizontal and vertical compositions follows easily from the definitions. ∎

A useful consequence of this construction will be that ! preserves certain adjunctions, namely those for which the unit and counit are monoidal natural. This situtuation will be called *monoidal adjunction* and denoted by $(F, \phi, \widehat{\phi}) \dashv (G, \psi, \widehat{\psi})$.

Corollary 4.7 *Let $(F, \phi, \widehat{\phi})\colon \mathcal{D} \to \mathcal{E}$ and $(G, \psi, \widehat{\psi})\colon \mathcal{E} \to \mathcal{D}$ be symmetric monoidal such that $(F, \phi, \widehat{\phi}) \dashv (G, \psi, \widehat{\psi})$. Then there exists an adjunction $F! \dashv G!$.*

Proof: η and ϵ are unit and counit of an adjunction $F \dashv G$ precisely when they satisfy the triangle inequalities. But then the 2-functoriality of ! ensures that $\eta!$, $\epsilon!$, $F!$ and $G!$ also satisfy the triangle inequalites. ∎

4.2 Completeness properties of $\mathcal{D}!$[1]

What properties of $\mathcal{D}$ ensure that $\mathcal{D}!$ is bicomplete? The existence of various limits and colimits in $\mathcal{D}!$ are here considered separately.

Firstly, the existence of coproducts can be assured by quite weak assumptions.

Proposition 4.8 *If $\mathcal{D}$ has an initial object 0 and a natural isomorphism $X \otimes 0 \cong 0$ then $\mathcal{D}!$ has all coproducts.*

Proof: The objects of the coproduct are the disjoint union of the objects of the summands. Within each summand homobjects, composition and identities are defined to be the same as in the summand. Homobjects between summands are defined to be 0 and the isomorphism

[1]The proof of theorem 4.12 is the author's. Other parts of this section are the results of collaborative work [13] and not claimed as entirely original.

between $X \otimes 0$ and 0 allows definition of composition. Checking that the category so defined is a coproduct is easy. ∎

Note that the conditions of the proposition are satisfied whenever $\mathcal{D}$ is closed.

Corollary 4.9 *Under the conditions of the preceding proposition the categories $\mathcal{D}! \downarrow$ **Set**, $\mathcal{D}!_\Sigma$ and $\mathcal{D}! \downarrow$ **Seti** have all small coproducts.*

Proof: By construction the forgetful functor $U_{\mathcal{D}!}$ preserves coproducts for any $\mathcal{D}$. Applying corollary 3.5 gives the result. ∎

The next two propositions are concerned with the existence of adjoints to the forgetful functor of $\mathcal{D}!$.

Proposition 4.10 *If $\mathcal{D}$ has an initial object 0 and $0 \otimes A \cong 0 \cong A \otimes 0$ then the forgetful functor $U_{\mathcal{D}!}: \mathcal{D}! \to \mathbf{Set}$ has a left adjoint $D: \mathbf{Set} \to \mathcal{D}!$.*

Proof: One can think of DS as the discrete $\mathcal{D}$-category on S, defined by $(DS)(X, X) = I$ and $(DS)(X, Y) = 0$ when $X \neq Y$. ∎

There is a dual result giving a clique (or codiscrete) category on any set.

Proposition 4.11 *If $\mathcal{D}$ has a final object 1 then $U_{\mathcal{D}!}: \mathcal{D}! \to \mathbf{Set}$ has a right adjoint $E: \mathbf{Set} \to \mathcal{D}!$.*

Proof: Set $(ES)(u, v) = 1$ for all $u, v \in S$. There is then a unique choice for the composition and identity morphisms. ∎

Theorem 4.12 *If $\mathcal{D}$ has all $\mathcal{J}$ limits (for small $\mathcal{J}$) then so does $\mathcal{D}!$.*

This result can be proved by constructing the desired limit from first principles. However, the proof is here split into two smaller parts, making it somewhat easier to understand.

For the first part recall that to say that $\mathcal{D}$ has all $\mathcal{J}$-limits means that the diagonal functor, denoted here by $\triangle: \mathcal{D} \to \mathcal{D}^{\mathcal{J}}$, has a right adjoint. For any category $\mathcal{J}$ and any monoidal $\mathcal{D}$, $\mathcal{D}^{\mathcal{J}}$ has an obvious "pointwise" monoidal structure. I show that the adjunction $\triangle \dashv \mathrm{Lim}$ lifts to an adjunction $\triangle! \dashv \mathrm{Lim}!$.

For the second part of the proof define (for any $\mathcal{D}$, not just those with $\mathcal{J}$-limits) a functor $\Psi: \mathcal{D}^{\mathcal{J}}! \to \mathcal{D}!^{\mathcal{J}}$ with the properties

i) $\Psi \circ (\triangle!)$ is the diagonal functor from $\mathcal{D}!$ to $\mathcal{D}!^{\mathcal{J}}$, and

ii) Ψ has a right adjoint provided $\mathcal{J}$ is small.

Since adjoints compose I have thus constructed a right adjoint to the diagonal functor from $\mathcal{D}!$ to $\mathcal{D}!^{\mathcal{J}}$. In other words, I will have shown that $\mathcal{D}!$ has all $\mathcal{J}$-limits.

In more detail, suppose $\mathcal{D}$ is a (symmetric) monoidal category $\mathcal{D}$ and $\mathcal{J}$ is any category. Define the tensor product on $\mathcal{D}^{\mathcal{J}}$ by

i) $(F \otimes G)J = FJ \otimes GJ$ and $(F \otimes G)f = (Ff) \otimes (Gf)$. Given natural transformations $\tau: F \to G$ and $\tau': F' \to G'$, define $\tau \otimes \tau'$ by setting $(\tau \otimes \tau')_J = \tau_J \otimes \tau'_J$.

ii) The tensor identity is the constant functor to the object I.

iii) Associativity, identity and symmetry morphisms are defined pointwise.

This makes $\mathcal{D}^{\mathcal{J}}$ a monoidal category, symmetric if $\mathcal{D}$ is symmetric.

Consider the diagonal functor $\triangle: \mathcal{D} \to \mathcal{D}^{\mathcal{J}}$. $\triangle$ is a strict monoidal functor, by definition of tensor product in $\mathcal{D}^{\mathcal{J}}$. If $\mathcal{D}$ has $\mathcal{J}$-limits then $\triangle$ has a right adjoint. By lemma 3.7 this right adjoint can be considered as a monoidal functor and, furthermore, the construction given in the proof of that lemma makes the adjunction a monoidal one. Apply corollary 4.7 to obtain:

Corollary 4.13 *The monoidal functor $\triangle!: \mathcal{D}! \to (\mathcal{D}^{\mathcal{J}})!$ has a right adjoint.*

To complete the proof that limits exist construct an adjoint pair between $\mathcal{D}^{\mathcal{J}}!$ and $\mathcal{D}!^{\mathcal{J}}$.

Define a functor $\Psi: \mathcal{D}^{\mathcal{J}}! \to \mathcal{D}!^{\mathcal{J}}$ as follows. For each $\mathcal{D}^{\mathcal{J}}$-category $\mathcal{A}$, $\Psi\mathcal{A}$ must be a functor from $\mathcal{J}$ to $\mathcal{D}!$. Define this functor on each object J of $\mathcal{J}$ to be a $\mathcal{D}$-category with

i) the same objects as $\mathcal{A}$;

ii) for each pair X and Y of objects define $(\Psi\mathcal{A}J)(X,Y) = (\mathcal{A}(X,Y))J$ (Since $\mathcal{A}$ is a $\mathcal{D}^{\mathcal{J}}$-category each $\mathcal{A}(X,Y)$ is a functor from $\mathcal{J}$ to $\mathcal{D}$. Evaluate this functor at J.);

iii) composition morphisms defined to be the J component of the composition in $\mathcal{A}$ (Composition morphisms in $\mathcal{A}$ are natural transformations; take the component of this transformation at J);

iv) similarly, the identity morphism $I \to \mathcal{A}(X,Y)J$ is the J component of the identity morphism in $\mathcal{A}$.

Given $f\colon J \to K$ in $\mathcal{J}$ define $(\Psi\mathcal{A})f$, a $\mathcal{D}$-functor, to be equality on objects (note that the objects of $(\Psi\mathcal{A})J$ and those of $(\Psi\mathcal{A})K$ are identical to the objects of $\mathcal{A}$). On homobjects define $((\Psi\mathcal{A})f)_{XY} = (\mathcal{A}(X,Y))f$.

Given a $\mathcal{D}^{\mathcal{J}}$-functor $F\colon \mathcal{A} \to \mathcal{B}$, ΨF must be a natural transformation. The J-component of ΨF must be a $\mathcal{D}$-functor. Define it by $(\Psi F)_J = F$ on objects, and

$$((\Psi F)_J)_{XY} = (F_{XY})J\colon \mathcal{A}(X,Y)J \to \mathcal{B}(FX,FY)J.$$

Proposition 4.14 *$(\Psi\mathcal{A})J$ is a well-defined $\mathcal{D}$-category for each J and $(\Psi\mathcal{A})f$ a well-defined $\mathcal{D}$-functor. For each $\mathcal{D}^{\mathcal{J}}$-functor $F\colon \mathcal{A} \to \mathcal{B}$ $(\Psi F)_J$ is a $\mathcal{D}$-functor and (ΨF) is natural in J. Furthermore Ψ preserves identities and composition. Hence it is a well-defined functor.*

Proof: Each statement is proved by showing that certain diagrams commute. In each case the required diagram can be seen as a special case of a diagram which commutes by the assumptions that $\mathcal{A}$ and $\mathcal{B}$ are $\mathcal{D}^{\mathcal{J}}$ categories, or by the assumption that F is a $\mathcal{D}^{\mathcal{J}}$-functor.

For example the diagram expressing the requirement that $(\Psi\mathcal{A})f$ preserves composition is this.

$$\begin{tikzcd}
((\Psi\mathcal{A})J)(X,Y) \otimes ((\Psi\mathcal{A})J)(Y,Z) \arrow[r, "\mu^{(\Psi\mathcal{A}J)}"] \arrow[d, "(\Psi\mathcal{A})f \otimes (\Psi\mathcal{A})f"'] & ((\Psi\mathcal{A})J)(X,Z)) \arrow[d, "(\Psi\mathcal{A})f"] \\
((\Psi\mathcal{A})K)(X,Y) \otimes ((\Psi\mathcal{A})K)(Y,Z) \arrow[r, "\mu^{(\Psi\mathcal{A}K)}"] & ((\Psi\mathcal{A})K)(X,Z))
\end{tikzcd}$$

Substituting definitions yields the diagram

$$\begin{tikzcd}
\mathcal{A}(X,Y)J \otimes \mathcal{A}(Y,Z)J \arrow[r, "\mu^{\mathcal{A}}_J"] \arrow[d, "\mathcal{A}(X,Y)f \otimes \mathcal{A}(Y,Z)f"'] & \mathcal{A}(X,Z)J \arrow[d, "\mathcal{A}(X,Z)f"] \\
\mathcal{A}(X,Y)K \otimes \mathcal{A}(Y,Z)K \arrow[r, "\mu^{\mathcal{A}}_K"] & \mathcal{A}(X,Z)K
\end{tikzcd}$$

But this is just part of the statement that $\mu^{\mathcal{A}}$ is a natural transformation.

Other parts of the proof are similar. ∎

Proposition 4.15 *The composite functor $\Psi(\triangle!): \mathcal{D}! \to \mathcal{D}!^{\mathcal{J}}$ is the diagonal functor.*

Proof: Immediate from definitions of $\triangle$ and Ψ. ∎

From now on assume that $\mathcal{J}$ is small. Construct a right adjoint to Ψ by giving a universal in $\mathcal{D}^{\mathcal{J}}!$ from Ψ to F for each functor $F: \mathcal{J} \to \mathcal{D}!$. Denote this object by ΦF and define it as follows.

i) Each object of ΦF is a mapping γ assigning an object of FJ to each J in $\mathcal{J}$, such that for any $f: J \to K$, $(Ff)(\gamma J) = \gamma K$. Alternatively, the object set of ΦF is the limit of the composite functor UF, where U is the forgetful functor from $\mathcal{D}!$ to **Set**.

ii) The homobject $(\Phi F)(\gamma, \delta)$ must be a functor $\mathcal{J} \to \mathcal{D}$. Define it to send each object J to $(FJ)(\gamma J, \delta J)$ and each arrow $f: J \to K$ to $(Ff)_{\gamma J, \delta J}$

iii) The composition map $\mu^{\Phi F}$ is to be a natural tranformation. Its component $\mu^{\Phi F}_{\gamma,\delta,\zeta} J$ must be a map

$$(FJ)(\gamma J, \delta J) \otimes (FJ)(\delta J, \zeta J) \to (FJ)(\gamma J, \zeta J)$$

Define this map to be a component of the composition in the $\mathcal{D}$-category FJ, namely $\mu^{FJ}_{\gamma J, \delta J, \zeta J}$.

iv) Similarly, the identity map $i^{\Phi F}$ must be a natural transformation $\triangle I \to (\Phi F)(\gamma, \gamma)$. Define it at each J by the identity map in FJ,

$$i^{FJ}_{\gamma J}: I \to (FJ)(\gamma J, \gamma J).$$

Proposition 4.16 *$(\Phi F)(\gamma, \delta)$ is a well-defined functor; $\mu^{\Phi F}$ and $i^{\Phi F}$ are natural transformations; and they satisfy the commutativity conditions required for ΦF to be a $\mathcal{D}^{\mathcal{J}}$-category.*

Proof: Once again this is a matter of diagram manipulation. ∎

Next define a map that will be the counit of the desired adjunction.

$\epsilon\colon \Psi\Phi F \to F$ is to be a natural transformation. Its component at each object J of $\mathcal{J}$ is to be a $\mathcal{D}$-functor. The objects of $(\Psi\Phi F)J$ are the objects of ΦF, by definition of Ψ. Thus they are cones to objects of F, by definition of Φ. Define $\epsilon^J\colon (\Psi\Phi F)J \to FJ$ on objects by

$$\epsilon^J(\gamma) = \gamma J.$$

For homobjects I require a map from $((\Psi\Phi F)J)(\gamma, \delta)$ to $(FJ)(\gamma J, \delta J)$. But we already know that $((\Psi\Phi F)J)(\gamma, \delta) = (\Phi F)(\gamma, \delta)J$ by definition of Ψ. This in turn is equal to $(FJ)(\gamma J, \delta J)$, by definition of Φ. The obvious definition is

$$\epsilon^J_{\gamma,\delta} = \text{identity map.}$$

Proposition 4.17 *ϵ^J is a $\mathcal{D}$-functor for each J and ϵ is natural in J.*

Proof: $\mathcal{D}$-functoriality is a standard argument and can easily be supplied by the interested reader. I will consider naturality in more detail. I require that this diagram of $\mathcal{D}$-functors commute:

$$\begin{tikzcd}
(\Psi\Phi F)J \arrow[r, "\epsilon^J"] \arrow[d, "(\Psi\Phi F)f"'] & FJ \arrow[d, "Ff"] \\
(\Psi\Phi F)K \arrow[r, "\epsilon^K"] & FK
\end{tikzcd}$$

First chase this diagram for objects. The objects of $(\Psi\Phi F)J$ are the objects of ΦF, which are cones to UF. In one direction I have

$$\begin{aligned}
(Ff)\epsilon^J\gamma &= (Ff)(\gamma J) && \text{(definition of } \epsilon) \\
&= \gamma K && \text{(property of } \gamma).
\end{aligned}$$

For the other direction recall that $(\Psi\mathcal{A})f$ is the identity map on objects. Hence

$$\begin{aligned}\epsilon^K((\Psi\Phi F)f)\gamma &= \epsilon^K\gamma \\ &= \gamma K \quad \text{(definition of } \epsilon).\end{aligned}$$

Next chase the diagram on homobjects. Recall that ϵ^J is the identity map on homobjects and that $((\Psi\Phi F)f)_{\gamma\delta}$ is $((\Phi F)(\gamma,\delta))f$, by definition of Ψ. Substituting the definition of Φ this is equal to $(Ff)_{\gamma,\delta}$, completing the proof. ∎

With the preceding results in hand I can now proceed to the proof of the theorem.

Proof of theorem 4.12: The final step is to show that Φ and ϵ define a universal. So suppose that $\tau\colon \Psi\mathcal{A} \to \mathcal{B}$ is an arrow of $\mathcal{D}!^{\mathcal{J}}$. I wish to show that there exists a unique arrow $\tau'\colon \mathcal{A} \to \Phi\mathcal{B}$ of $\mathcal{D}^{\mathcal{J}}!$ such that τ factors as

$$\Psi\mathcal{A} \xrightarrow{\Psi\tau'} \Psi\Phi F \xrightarrow{\epsilon} F.$$

Consider uniqueness first. $\tau = \epsilon(\Psi\tau')$ if and only if $\tau_J = \epsilon^J(\Psi\tau')_J$ for each J. These two expressions must agree on both objects and homobjects of $(\Psi\mathcal{A})J$. The objects of $(\Psi\mathcal{A})J$ are exactly the objects of $\mathcal{A}$ (by definition of Ψ) so for each X

$$\begin{aligned}\tau_J X &= \epsilon^J(\Psi\tau')_J X \\ &= \epsilon^J\tau' X \quad \text{(definition of } \Psi) \\ &= (\tau' X)J \quad \text{(definition of } \epsilon).\end{aligned}$$

Note that this defines the action of τ' on objects of $\mathcal{A}$ uniquely.

Also, for a pair of objects X and Y I must show

$$\begin{aligned}(\tau_J)_{XY} &= (\epsilon^J)_{\Psi\tau'_J X, \Psi\tau'_J Y}((\Psi\tau')_J)_{XY} \\ &= ((\Psi\tau')_J)_{XY} \quad \text{(definition of } \epsilon) \\ &= (\tau'_{XY})_J \quad \text{(definition of } \Psi).\end{aligned}$$

Existence is a matter of showing that τ' so defined is a map of $\mathcal{D}^{\mathcal{J}}!$. That is, I must show that $\tau' X$ is a well-defined object of ΦF; that τ'_{XY} is a natural transformation from the functor $\mathcal{A}(X,Y)$ to $(\Phi F)(\tau' X, \tau' Y)$; and that τ' preserves composition and identities. These results are consequences of each τ_J being a $\mathcal{D}$-functor. ∎

The situation for general colimits is more complex. As we have already seen, coproducts can be constructed in $\mathcal{D}!$ under weak assumptions on $\mathcal{D}$. In fact $\mathcal{D}$ need not have coproducts itself. However to construct coequalizers in $\mathcal{D}!$ is difficult and, in general, requires the assumption that $\mathcal{D}$ has *all* colimits. Hence we must settle for a coarser result than theorem 4.12.

Theorem 4.18 *If $\mathcal{D}$ is cocomplete and tensor product distributes over small coproducts, that is*

$$X \otimes (\sum_{\alpha} Y_\alpha) \cong \sum_{\alpha} (X \otimes Y_\alpha).$$

then $\mathcal{D}!$ is cocomplete.

Proof: See Betti et al [6]. ∎

Observe, once again, that the conditions of the theorem are satisfied whenever $\mathcal{D}$ is closed.

Theorem 4.19 *If $\mathcal{D}$ is symmetric, complete and closed then $\mathcal{D}!$ is closed.*

Outline of Proof: (A full proof is given by Kelly [29, page 55]; this outline is essentially Lawvere's [30, page 153].) The essential idea is that, given $\mathcal{D}$-categories $\mathcal{A}$ and $\mathcal{B}$, the objects of the exponential $[\mathcal{A}, \mathcal{B}]$ are the $\mathcal{D}$-functors from $\mathcal{A}$ to $\mathcal{B}$. The homset $[\mathcal{A}, \mathcal{B}](F, G)$ is defined in a manner analogous to the definition of the set of natural transformations between two ordinary functors. Ignoring for the moment the naturality condition, a transformation between ordinary functors F and G is a family of arrows, and the set of all such transformations is the following product. ($\mathrm{ob}(\mathcal{A})$ is the set of objects of the category $\mathcal{A}$.)

$$P_1 = \prod_{X \in \mathrm{ob}(\mathcal{A})} \mathcal{B}(FX, GX).$$

Now the natural transformations are those elements of of P_1 which satisfy certain equations. The trick is to express the satisfaction of the equations as an equalizer.

To do this, suppose that τ is a transformation in P_1 and $f: Y \to Z$ is an arrow of $\mathcal{A}$. Consider the expression $(Gf)\tau_Y$. This is actually a function of f and τ, and can be expressed as the following composite map in **Set**:

$$P_1 \times \mathcal{A}(Y, Z) \xrightarrow{\pi_Y \times G_{YZ}} \mathcal{B}(FY, GY) \times \mathcal{B}(GY, GZ) \xrightarrow{\mu} \mathcal{B}(FY, GZ).$$

Taking the adjoint transpose of this map

$$p_{YZ}: P_1 \to [\mathcal{A}(Y, Z), \mathcal{B}(FY, GZ)].$$

effectively quantifies over all $f: Y \to Z$.

Similarly, the expression $\tau_Z(Ff)$ can be expressed as the composite map (this time we must insert the interchange morphism t)

$$P_1 \times \mathcal{A}(Y,Z) \xrightarrow{\pi_Z \times F} \mathcal{B}(FZ,GZ) \times \mathcal{B}(FY,FZ) \xrightarrow{\mu t} \mathcal{B}(FY,GZ),$$

and taking the adjoint transpose gives a second arrow

$$q_{YZ}: P_1 \to [\mathcal{A}(Y,Z), \mathcal{B}(FY,GZ)].$$

Now τ is natural precisely when $p(\tau) = q(\tau)$ for all choices of Y and Z. But we can now quantify over Y and Z by taking a product, defining P_2 by

$$P_2 = \prod_{Y,Z \in \mathrm{ob}(\mathcal{A})} [\mathcal{A}(Y,Z), \mathcal{B}(FY,GZ)]$$

and defining the natural transformations to be the equalizer of the maps

$$p, q: P_1 \to P_2.$$

But this has defined the set of natural transformations using only products, equalizers and homobjects in **Set**. The same definition can be transferred to any complete, closed $\mathcal{D}$. ∎

Restating this theorem in slightly different terms: there exists an object of $\mathcal{D}$ that can be seen as the *object of $\mathcal{D}$-natural transformations* from a given $\mathcal{D}$-functor to a parallel one. $\mathcal{D}$-natural transformations can also be defined individually, as the arrows from I to an object of $\mathcal{D}$-natural transformations.

In the case that $\mathcal{D}$ is a preorder any pair of parallel arrows in $\mathcal{D}$ are equal, so the equalizer of p and q is P_1 itself. Since products in a preorder are simply greatest lower bounds it follows that

$$[F,G] = \inf_X [FX, GX].$$

Moreover, there is at most one $\mathcal{D}$-natural transformation (as defined in the preceding paragraph) from F to G, so the set of $\mathcal{D}$-functors from $\mathcal{A}$ to $\mathcal{B}$ can also be seen as a preorder with $F \leq G$ whenever $[FX, GX] \geq I$ for all objects X of $\mathcal{A}$. For example, if $\mathcal{D}$ is **2**, so that $\mathcal{D}$-functors are monotone maps, $F \leq G$ means that F is less than G in the usual pointwise ordering.

4.3 Additional constructions of MC categories

In this section I consider some constructions of MC categories other than the use of enrichment and labeling. The first group of results concerns subcategories of SSM categories. Examples of these include the category of posets (which is a subcategory of **2**!) and the category of metric spaces (a subcategory of $\mathbf{R}_+^{\mathrm{op}}!$).

4.3.1 Subcategories of SSM categories

Here I consider conditions under which subcategories of SSM categories are closed. Here are two such conditions which are quite simple to check.

Lemma 4.20 *Let $\mathcal{D}$ be a closed SSM category and $\mathcal{E}$ a full sub-SSM-category of $\mathcal{D}$. Suppose that for all objects X and Y in $\mathcal{E}$, $[X, Y]$ is in $\mathcal{E}$. Then $\mathcal{E}$ is closed.*

Proof: $f: Z \to [X, Y]$ is an arrow of $\mathcal{E}$ if and only if f is an arrow of $\mathcal{D}$ if and only if there exists a corresponding arrow $f': Z \otimes X \to Y$ in $\mathcal{D}$ if and only if $f': Z \otimes X \to Y$ exists in $\mathcal{E}$. ∎

Corollary 4.21 **Pos**, *the category of posets, is (cartesian) closed.*

Proof: **2**!, the category of preordered sets, is closed, by theorem 4.19. **Pos** is a full subcategory satisfying the conditions of the lemma ∎

A second condition under which subcategories are closed is given by the next lemma.

Lemma 4.22 *Let $\mathcal{D}$ be a closed SSM category and $\mathcal{E}$ be a full sub-SSM-category that is coreflective in $\mathcal{D}$ (that is, the inclusion functor has a right adjoint). Then $\mathcal{E}$ is closed.*

Proof:

Suppose that the right adjoint of inclusion is $R: \mathcal{D} \to \mathcal{E}$. Then for X, Y, Z objects of $\mathcal{E}$, $\hom_{\mathcal{E}}(X \otimes Y, Z)$ is equal to $\hom_{\mathcal{D}}(X \otimes Y, Z)$ which is naturally isomorphic to $\hom_{\mathcal{D}}(X, [Y, Z])$ which is naturally isomorphic to $\hom_{\mathcal{E}}(X, R([Y, Z]))$. Hence the right adjoint of $- \otimes Y$ is $R([Y, -])$ and $\mathcal{E}$ is closed. ∎

4.3.2 Finite monoidal posets

The following are some general theorems about finite ordinals which are given a monoidal structure. We have seen several important examples of such categories, namely the categories **2**, **3** and **3**$'$.

First consider how such categories might be combined using $\rhd$. This means that forgetful functors must be defined for them. The following theorem shows that there are only limited possibilites for such a forgetful functor.

Theorem 4.23 *Suppose n is a finite ordinal equipped with an idempotent monoidal structure and $U: n \to$ **Set** is a strong monoidal functor. Then for every object a in n either $Ua = 0$ or Ua is a singleton.*

Proof: Suppose not and let a be an element of n such that Ua is neither empty nor a singleton. Now U is strong monoidal and $\otimes$ is idempotent, so $Ua \times Ua$ must be isomorphic to Ua. Hence, since Ua is not empty or a singleton it must be infinite.

Now either $I < a$ or $I > a$ (I is the tensor identity of n). Consider these two cases separately:

1) $I < a$. Denote by j the arrow from I to a. Naturality of the isomorphism between $U(a \otimes b)$ and $Ua \times Ub$ requires that the following commute:

$$\begin{tikzcd} UI \times Ua \arrow[r, leftrightarrow, "l"] \arrow[d, "Uj \times 1"'] & Ua \arrow[d, equal] \\ Ua \times Ua \arrow[r, leftrightarrow] & Ua \end{tikzcd}$$

But UI is a singleton and Ua is infinite, so $Uj: UI \to Ua$ cannot be surjective. But then $Uj \times 1$ cannot be surjective, so that the composite

$$Ua \to UI \times Ua \to Ua \times Ua \to Ua$$

cannot be surjective, so it certainly cannot be the identity. Hence the diagram does not commute.

2) $a > I$. Denote the unique map $a \to I$ by j. Naturality would require commutativity of

$$
\begin{tikzcd}
Ua \times Ua \arrow[r, leftrightarrow, "l"] \arrow[d, "Uj \times 1"'] & Ua \arrow[d, equal] \\
UI \times Ua \arrow[r, leftrightarrow] & Ua
\end{tikzcd}
$$

But Uj here cannot be injective so a similar argument to case 1 shows that the diagram cannot commute because the composite is not injective.

This completes the proof. ∎

The following lemma establishes a necessary and sufficient condition for simple categories (such as the finite ordinals under consideration) to be closed.

Lemma 4.24 *Suppose L is a cocomplete poset with tensor product. Then L is closed if and only if tensor product distributes over all sums.*

Proof: Given elements b and c define $[b, c]$ by

$$[b, c] = \sup\{\, x \mid x \otimes b \leq c \,\}$$

Now for any $a \leq [b, c]$

$$
\begin{aligned}
a \otimes b &\leq [b, c] \otimes b \\
&= \sup\{\, x \otimes b \mid x \otimes b \leq c \,\} \quad (\otimes \text{ distributes over sup}) \\
&\leq c.
\end{aligned}
$$

Conversely if $a \otimes b \leq c$ then $a \leq [b, c]$ by definition of $[b, c]$. ∎

Now consider finite ordinals n and m each equipped with a closed idempotent tensor product. By theorem 4.23 there is at most one map in **Set** from $U_n(x)$ to $U_m(y)$ for any $x \in n$ and $y \in m$. Hence the comma category $n \rhd m$ is easily seen to be the subcategory of $n \times m$ omitting those pairs $\langle x, y \rangle$ for which $U_n(x) = 1$ and $U_m(y) = 0$.

Proposition 4.25 *Coproduct in $n \rhd m$ can be calculated pointwise.*

Proof: I need only show that if $\langle x, y \rangle$ and $\langle x', y' \rangle$ satisfy the condition so does $\langle x + x', y + y' \rangle$. Observe that $x + x' = \max\{x, x'\}$. So if $U_m(y + y') = 0$ then $U_m(y) = U_m(y') = 0$. Hence, by assumption, $U_n(x) = U_n(x') = 0$. But $U_n(x + x')$ equals either $U_n(x)$ or $U_n(x')$ and therefore is also zero and the proof is done. ∎

Now, by definition, tensor product in $n \rhd m$ is also calculated pointwise. If the tensor products on n and m are both closed then they distribute over coproducts by the lemma. But then tensor product in $n \rhd m$ distributes over coproduct in $n \rhd m$. Apply the lemma in the opposite direction to obtain:

Theorem 4.26 *If n and m are finite ordinals with closed idempotent tensor products then $n \rhd m$ is also closed monoidal.*

Observe that this result fails even for very simple cases of infinite ordinals. For consider $\omega + 1$. This is cartesian closed (any chain with maximal element is cartesian closed). Consider a functor to **Set** sending all finite elements to the empty set and ω to a singleton. With this forgetful functor $(\omega + 1) \rhd (\omega + 1)$ can be described as containing those pairs $\langle a, b\rangle$ where a and b are either integers or infinity and such that if $a = \infty$ then $b = \infty$.

But $(\omega+1) \rhd (\omega+1)$ is not cartesian closed. For suppose that the internal-hom functor exists, and suppose that the exponential $[\langle 0, 1\rangle, \langle 0, 0\rangle] = \langle a, b\rangle$ for some a and b. Now observe that for any n

$$\langle n, 0\rangle \times \langle 0, 1\rangle = \langle \min(n, 0), \min(0, 1)\rangle \leq \langle 0, 0\rangle,$$

by an easy propery of product in this category. Taking the adjoint transpose shows that for any value of n

$$\langle n, 0\rangle \leq [\langle 0, 1\rangle, \langle 0, 0\rangle] = \langle a, b\rangle.$$

Thus a must be ∞, so $b = \infty$ too. But to claim that $[\langle 0, 1\rangle, \langle 0, 0\rangle] = \langle \infty, \infty\rangle$ is equivalent to claiming that for all objects $\langle x, y\rangle$,

$$\langle x, y\rangle \leq [\langle 0, 1\rangle, \langle 0, 0\rangle],$$

or equivalently that for all $\langle x, y\rangle$

$$\langle \min(x, 0), \min(y, 1)\rangle = \langle x, y\rangle \times \langle 0, 1\rangle \leq \langle 0, 0\rangle.$$

But this is clearly absurd and contradicts the assumption that the internal-hom exists.

4.4 Iterating ! and $\triangleright$

At the start of chapter 3 I observed that it is frequently tedious to show that a given category is MC or MCE. The $\triangleright$ and ! constructions were introduced to simplify such proofs, the intention being that, under certain conditions, expressions built up by repeated use of $\triangleright$ and ! could be easily recognized as defining an MC or MCE category.

The results of the last two chapters give such conditions, but they are quite complex and it would be convenient if they could be simplified into one or two rules that cover a reasonably large class of expressions. I now build on the simple rule given in section 3.6, where I analyzed expressions by the existence of adjoints to forgetful functors.

Restrict attention to categories that are bicomplete, symmetric monoidal and closed. I wish to define a class of expressions in such categories, ! and $\triangleright$ that denote categories with the same set of properties.

Firstly, if $\mathcal{D}$ has the properties then $D!$ is strong symmetric monoidal by lemma 4.1. Theorems 4.12 and 4.18 ensure that $\mathcal{D}!$ is bicomplete, and propositions 4.10 and 4.11 show that it has both adjoints. So a category of the form $\mathcal{D}!$ is well-behaved enough to appear as the argument to ! or as either argument of $\triangleright$.

In section 3.6 I gave conditions under which the forgetful functor of $\mathcal{D} \triangleright \mathcal{E}$ has a right adjoint. I now consider its left adjoint.

Lemma 4.27 *Let $F: \mathcal{A} \to \mathcal{C}$ and $G: \mathcal{B} \to \mathcal{C}$ be functors having left adjoints F' and G', respectively. Suppose further that FF' is the identity functor on $\mathcal{A}$ (so that $\mathcal{C}$ can be seen as a coreflective subcategory of $\mathcal{A}$.) Then the projection functor $(F \downarrow G) \xrightarrow{\pi} \mathcal{A}$ has a left adjoint.*

Proof: Let $\eta^G: C \to GG'C$ be the unit of the adjunction $G' \dashv G$. I claim that the "free $(F \downarrow G)$ object on an object C" can be defined by

$$FF'C = C \xrightarrow{\eta^G} GG'C,$$

and the unit of this adjunction is the unit of the adjunction $F' \dashv F$, denoted $\eta^F: C \to FF'C = \pi\langle F'C, \eta^G, G'C\rangle$.[2]

[2] $FF'C = C$ but it is not necessary that η^F be the identity arrow of C, although it is always an automorphism.

For suppose that $p: C \to \pi\langle A, f, B\rangle$ is an arrow of $\mathcal{C}$. By definition of π, the codomain of p is FA, hence p corresponds to an arrow

$$F'C \xrightarrow{p^\flat} A.$$

The adjunction $G' \dashv G$ then ensures the existence of a unique q making the following diagram commute

$$\begin{tikzcd} C \arrow[r, "\eta^F"] \arrow[dr, "p"'] & FF'C \arrow[r, "\eta^G"] \arrow[d, "Fp^\flat"] & GG'C \arrow[d, dotted, "Gq"] \\ & FA \arrow[r, "f"'] & GB \end{tikzcd}$$

Thus $\langle p^\flat, q\rangle$ is an arrow factoring p. Properties of the two adjunctions ensure that it is the unique such arrow. This completes the proof. ∎

Corollary 4.28 *Let $\mathcal{D}$ and $\mathcal{E}$ be objects of* **SSM**, *and suppose that $\mathcal{D}$ has an initial object 0 preserved by tensor product with any object. Let $U_{\mathcal{E}}$ have a left adjoint. Then the forgetful functor from $\mathcal{D}! \rhd \mathcal{E}$ also has a left adjoint.*

Proof: By proposition 4.10 $U_{\mathcal{D}!}$ has a left adjoint D taking any set X to the discrete category DX. Note that $U_{\mathcal{D}!}DX = X$. So the lemma applies, and the projection $\mathcal{D} \rhd \mathcal{E} \xrightarrow{\pi} \mathcal{D}$ has a left adjoint. By composition with D a left adjoint to $U_{\mathcal{D}\rhd\mathcal{E}}$ is obtained. ∎

Suppose that E is an expression formed by repeated application of ! and $\rhd$ to symmetric monoidal closed bicomplete categories $\mathcal{A}_1 \ldots \mathcal{A}_n$. I wish to ensure that the category defined by E is also symmetric monoidal closed and bicomplete. Since any subexpression of the form $E_1!$ is well behaved, I need only be concerned with subexpressions of the form $E_1 \rhd E_2$. It is sufficient to ensure that E_1 has a right adjoint for its forgetful functor and that E_2 has a left adjoint for its forgetful functor. This can be accomplished by the following syntactic conditions on E:

- Each $\mathcal{A}_i$ appearing as the left argument of $\rhd$ has a right adjoint for its forgetful functor;

- Each $\mathcal{A}_i$ appearing as the right argument of $\triangleright$ has a left adjoint for its forgetful functor;
- If the right argument of $\triangleright$ is of the form $E_1 \triangleright E_2$ then $E_1 = E_3!$ for some E_3.

Such conditions are sufficient to show that a fairly wide range of expressions denote MCE categories. This includes, for example, all of the expressions given in the table at the beginning of chapter 3.

Chapter 5

Future Research

In this chapter I list several possibilities for further investigation.

5.1 Schedules, automata and exponentiation

Suppose a preordered set $\mathcal{A}$ is viewed as a schedule, so its elements are seen as events or tasks. Each monotone map $F\colon \mathcal{A} \to \mathbf{2}$ can be seen as a stage of execution of $\mathcal{A}$ by considering those tasks mapped to 0 to be completed and those mapped to 1 to be still in progress. A second monotone map G can be seen as a later (i.e. more completed) stage of execution than F provided that $G \leq F$, because this relation holds only when G maps each event to a later stage than F. The category of **Bool**-functors $[\mathcal{A}, \mathbf{2}]$ can therefore be seen as an automaton for executing the schedule $\mathcal{A}$. Its initial state is the constant map to 1, its final state is the constant map to 0 and a transition exists from F to G if and only if $G \leq F$.

I discuss below how this view might be extended to some other monoidal categories. In each case the exponential $[\mathcal{A}, \mathcal{D}]$ has an interpretation as an automaton that executes the tasks of schedule $\mathcal{A}$ in accordance with the constraints of $\mathcal{A}$.

5.1.1 $\mathbf{3}'$-schedules and automata

To describe $[\mathcal{A}, \mathbf{3}']$, $\mathbf{3}'$ must first be described as a $3'$-category. As noted in section 2.1, $\mathbf{3}'(x, y)$ is by definition the internal-hom $[x, y]$. $[x, y]$ is defined to be the largest object z such that $x \otimes z \leq y$. From table 5.1 for tensor product in $\mathbf{3}'$ one can derive that $[0, x] = 2$ for all x, $[1, 0] = 0$, $[1, 1] = 1$ and so on.

	0	1	2
0	0	0	0
1	0	1	2
2	0	2	2

Table 5.1: Tensor product in $\mathbf{3}'$

Under the usual interpretation, in which $[x, y] = 1$ means $x \leq y$ and $[x, y] = 2$ means $x < y$, $3'$ can be described as a $3'$-category by the following diagram.

$$\begin{tikzcd} 0 \arrow[loop above, "<"] \arrow[r, "<"] & 1 \arrow[loop above, "\leq"] \arrow[r, "<"] & 2 \arrow[loop above, "<"] \end{tikzcd}$$

(The relation $0 < 2$ is omitted from the diagram. All other pairs of objects are unrelated.)

Suppose that A and B are objects of a schedule $\mathcal{A}$. For $F\colon \mathcal{A} \to \mathbf{3}'$ to be a $\mathbf{3}'$-functor FA and FB must be more strongly constrained in $\mathbf{3}'$ than A and B are in $\mathcal{A}$. By referring to the diagram we see that if $A \leq B$ then FA cannot be a smaller number than FB. Moreover, if $A < B$ then FA cannot be a smaller number than FB, nor can $FA = FB = 1$.

These requirements can be interpreted as follows. Take $FA = 0$ to mean that A has been completed, $FA = 1$ to mean A is in progress and $FA = 2$ to mean A has not been started. Then we can interpret $A \leq B$ to mean "B cannot begin before A begins, and cannot complete before A completes;" and $A < B$ to mean "B cannot begin until A has completed." These interpretations are completely consistent with the original motivation for defining $\mathbf{3}'$.

Consider now the object of natural transformations $[G, F]$, recalling from the remarks following theorem 4.19 that it is defined as $\inf_A [GA, FA]$.

For **Bool**-functors $G \leq F$ could be interpreted as G being a later stage of execution than F. In the case of $3'$-functors the situation is slightly more complex, because the homobject $[G, F]$ and the ordering on $\mathbf{3}'$-functors (which was also defined after theorem 4.19) are now different.

Recall that F is no greater than G if $[FX, GX]$ is never less than 1 for any X. This reduces to the statement that F must send each X to a later stage of execution than does G, so the same interpretation of the ordering of F and G applies here as for **Bool**-functors.

But $[G, F]$ contains further information about the delay from state F to state G. Recall that $[G, F] = \inf_A [GA, FA]$. Thus the delay from F to G is 0 if any event A is in an earlier state in G than in F, that is if G does not represent a later stage than F. Otherwise the delay is 1 only if there exists an event A for which $GA = FA = 1$. This can be restated as follows. It may be impossible for state G to follow state F, and in this case $[G, F] = 0$. Otherwise two cases arise: when there exists an event which is in progress in both F and G then G weakly follows F; otherwise G strongly follows F.

5.1.2 3-schedules and automata

A similar analysis can be performed for **3**, where the object 1 is thought of as the constraint $\leq^t$ (an accidental temporal constraint), and 2 as $\leq^c$ (a causal constraint).

Considered as a **3**-category, **3** is this:

$$\begin{tikzcd} 0 \arrow[loop above, "c"] \arrow[r, "c"] & 1 \arrow[loop above, "c"] \arrow[r, shift left, "c"] & 2 \arrow[loop above, "c"] \arrow[l, shift left, "t"] \end{tikzcd}$$

(The remaining relations are $0 \leq^c 2$, $1 \not\leq 0$ and $2 \not\leq 0$.)

Several interpretations of the objects as states are possible; the one used here is that 0, 1 and 2 are "completed," "active" and "idle." Under this interpretation $A \leq^t B$ means that B cannot be completed until A is completed, while $A \leq^c B$ means that if B is active then A must also be active (and if B is complete so must A be complete.) Once again, this is consistent with the original motivation for defining **3**.

If $[G, F] = 2$ then every task is at a later stage in F than in G. $[G, F] = 0$ arises only if there is a task A with $FA = 0$ and $GA \neq 0$. $[G, F] = 1$ holds when there exists a task A with $FA = 1$ and $GA = 2$, so that A has moved from "active" back to "idle." One can imagine the execution of a **3**-schedule as alternating each task between active and idle states until it is finally completed. A transition from F to G where $[G, F] = 1$ is an allowable backward move, resulting in some tasks that were active in F becoming idle in G. When $[G, F] = 2$ the transition is a forward move.

5.1.3 Real-valued schedules and automata

For $\mathbf{R}_+$, consider ∞ to be the state in which execution has not yet begun and $-\infty$ the completed state. Other objects represent incomplete task; the numeric value can be seen as the maximum amount of time required to complete an event. So the constraint $\mathcal{A}(A, B) = x$ says that if B will be completed y units hence then A must be completed after $y - x$ units. (In particular, if $y < x$ and B will be completed within x units then A must already be completed.)

The distance from state F to state G is $\inf_A(FA - GA)$. Thus if state G follows F by x units then every event has aged by at least x units.

For $\mathbf{R}_+^{\mathrm{op}}$, each state is the length of time since an action began executing. Thus 0 is the state in which no work has been done on an action and ∞ can be seen as the state where each action has been completed. It is straightforward to see that $\mathcal{A}(A, B) = x$ allows any amount of work to be performed on A without doing anything on B but once B has begun then A must follow at a distance of at most x units.

In this case the distance from F to G is $\sup_A(FA - GA)$, so that if G follows F by x units then every action has progressed at most an additional x units.

A similar explanation of exponentiation in $\mathbf{SR}^+!$ is a somewhat less obvious. One possiblity is to consider a number of stations that broadcast messages to all other stations over a communication network. Whenever a station receives a message on an incoming channel it forwards it on all outgoing channels. This situation can be represented by an $\mathbf{SR}^+$-category $\mathcal{A}$ in which the objects are the stations and a homobject $\mathcal{A}(X, Y)$ represents the set of possible transmission times between station X and station Y.

A functor from $\mathcal{A}$ to $\mathbf{SR}^+$ can be seen as assigning to each site the set of times when it may need to communicate. In the initial stage each site must be ready to communicate at all times. In the final configuration no site need communicate at all. (So each site has completed its task).

Intuitively speaking, there are two ways to progress from one state to another in this model. We can accept the existing schedules as they are, adhering to them as time passes. This amounts to subtracting a constant from every element of the state and discarding those that become negative. Or we can refine the schedule by agreeing among the sites that a previously scheduled communication will not be necessary. This amounts to taking a subset of the existing schedules.

This intuition is formalized in the definition of distance from stage F to G. Formally, this distance is an object of $\mathbf{SR}^+$, defined by $x \in [G, F]$ if and only if $\forall A \; GA + x \subseteq FA$. Informally speaking, x is an element of the distance from F to G if, by beginning in state F, waiting x units and then deleting certain times from the schedule we can obtain state

G. In other words x is in the distance from state F to state G if G could occur precisely x units after an occurrence of F in the normal operation of the network.

5.1.4 Conclusion

The upshot of these observations is that exponentiation can be seen as an operation taking a schedule and yielding an abstract machine or automaton for executing the events of the schedule. This state of affairs is strongly reminiscent of Stone duality (see Johnstone [26]). It would be very interesting to investigate this connection further.

5.2 Choice

Although the structures studied in this thesis were suggested by application to the semantics of concurrency, I have presented only a part of the required theory. A full model of concurrency must incorporate a notion of choice between different schedules.

Pratt did this in his original pomset model by modeling a process as a set of pomsets. However treating a process simply as a set of objects of a given MCE-category is not adequate. For suppose we wish to specify that an action may completely fail to occur with a certain probability. (An example would be a noisy channel that sometimes loses a message.) We are forced to include two schedules in the process, one containing the action and one not containing it. But if processes are simply sets without additional structure then there is no way to specify the probability of the action occurring.

One approach to this problem would be to model processes by an enriched category analogue of indexed category, and this is currently under investigation by R. Crew.

5.3 Application to programming languages

Once a reasonable method for introducing choice is found, one could consider the definition of programming languages based on this model. The outlines of such a language are already visible; a program would be a specification of an entire process, not just the program to be followed by one processor in a network. A process would be built up from simpler processes by various operations corresponding to the constructions presented in this thesis and the additional ones needed for choice. A compiler could separate out the activities of each processor from the overall specification. No doubt this project would suggest important changes to the theoretical framework.

5.4 Relation to other theories

Better understanding of the relationship between this theory and other theories of concurrency is needed. Consider R. Milner's calculus for communicating systems (CCS) [36, 37]. To bring CCS and the pomset model into closer correspondence, a CCS term might be considered as a language (that is, a set of totally ordered pomsets) on an alphabet of "internal actions" of the CCS term, together with a map from this alphabet to the external actions of the CCS term. Several distinct internal actions may correspond to a single external action. During execution, an agent's state can be represented by the sequence of internal actions that it has performed. It is possible for a single sequence of the external actions to correspond to several different sequences of internal actions. Thus it is possible for an agent to be in one of several states after it performs a single sequence of actions.

Suppose **Alph** were a suitable category for representing CCS alphabets (such as sets with a distinguished element τ and a complementation operation) and let **Lin** be the category of linear ordered sets. Then I could consider each word of the language representing a CCS term to be an object of the category $\mathbf{Lin} \triangleright (\mathbf{Set} \triangleright \mathbf{Alph})$. (Though this abuses notation, since neither **Lin** nor **Alph** have been equipped with a monoidal structure or a forgetful functor.) With this correspondence it would be interesting to see how the operations correspond to CCS terms, and also how CCS concepts such as bisimulation appear in the framework presented in this dissertation.

5.5 2-categorical framework

Just as **Cat** is not merely a category but a 2-category, $\mathcal{D}!$ is not merely a category but a $\mathcal{D}!$-category. Furthermore, in the same way that the notion of limit can be expanded to indexed or lax limits in 2-categories, so can it be expanded to a corresponding notion of indexed limits in $\mathcal{D}!$-categories. The concatenation operation appears as the $\mathcal{D}!$-categorical version of the opcomma construction. It may be possible to recast all the definitions of the operations of the schedules as such indexed limits and colimits, and this may well be more natural than the present definitions.

5.6 Change of basis category

Set has a special status as the codomain of all forgetful functors. This may rule out certain applications, even though many of the theorems are valid for other possible choices such as **Cat** or the category of sets and partial functions. A method of parametrizing the base category may be necessary.

Bibliography

[1] Gul A. Agha. *Actors: A Model Of Concurrent Computation in Distributed Systems.* MIT Press Series In Artificial Intelligence. MIT Press, 1986.

[2] Michael A. Arbib and Ernest G. Manes. *Arrows, Structures, and Functors: The Categorical Imperative.* Academic Press, New York, 1975.

[3] David B. Benson. Bialgebras: Some foundations for distributed and concurrent computation. Technical Report CS-87-168, Computer Science Department, Washington State University, Pullman, Washington, 1987.

[4] David B. Benson. Bisimulation of automata. Technical Report CS-87-167, Computer Science Department, Washington State University, Pullman, Washington, May 1987. Revision of work presented at LICS 1986.

[5] David B. Benson. The shuffle bialgebra. Technical Report CS-87-168, Computer Science Department, Washington State University, Pullman, Washington, May 1987.

[6] R. Betti, A. Carboni, R. Street, and R. Walters. Variation through enrichment. *Journal of Pure and Applied Algebra*, 29:109–127, 1983.

[7] Garrett Birkhoff. An extended arithmetic. *Duke Mathematical Journal*, 3(2):311–316, June 1937.

[8] Garrett Birkhoff. Generalized arithmetic. *Duke Mathematical Journal*, 9(2):283–302, June 1942.

[9] J. Dean Brock and William B. Ackerman. Scenarios: A model of non-determinate computation. In J. Díaz and I. Ramos, editors, *Proc. International Colloquium on*

the Formalization of Programming Concepts, Lecture Notes in Computer Science 107, pages 252–259, Peñíscola, Spain, April 1981. Springer-Verlag.

[10] S. D. Brookes, C. A. R. Hoare, and A. W. Roscoe. A theory of communicating sequential processes. *Journal of the ACM*, 31(3):560–599, July 1984.

[11] Stephen D. Brookes. *A Model for Communicating Sequential Processes*. D. Phil. thesis, University College, Oxford University, 1983. Also available as Carnegie-Mellon University Computer Science Department technical report CMU-CS-83-149.

[12] Manfred Broy, editor. *Control Flow and Data Flow: Concepts of Distributed Programming*, volume 14 of *NATO Advanced Science Institute Series F*. Springer-Verlag, 1985.

[13] Ross Casley, Roger F. Crew, José Meseguer, and Vaughan Pratt. Temporal structures. Technical Report STAN-CS-89-1297, Stanford University, 1989. (Revised version of a paper that appeared in LNCS 389.).

[14] Pierpaolo Degano, José Meseguer, and Ugo Montanari. Axiomatizing the algebra of net computation and processes. Technical Report SRI-CSL-90-12, Computer Science Laboratory, SRI International, November 1990.

[15] Pierpaolo Degano and Ugo Montanari. Distributed systems, partial orderings of events and event structures. In Broy [12], pages 7–106.

[16] Pierpaolo Degano and Ugo Montanari. Concurrent histories: A basis for observing distributed systems. *Journal of Computer and System Sciences*, 34(2/3):422–461, April/June 1987.

[17] Samuel Eilenberg and G. Max Kelly. Closed categories. In S. Eilenberg, D. K. Harrison, S. MacLane, and H. Röhrl, editors, *Proceedings of the Conference on Categorical Algebra; La Jolla 1965*, pages 421–562. Springer-Verlag, 1966.

[18] R.W. Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.

[19] H. Gaifman. Modeling concurrency by partial orders and nonlinear transition systems. In *Proc. REX School/Workshop on Linear Time, Branching Time and Partial*

Order in Logics and Models for Concurrency, pages 467–488, Noordwijkerhout, The Netherlands, May 1988. Springer-Verlag.

[20] Haim Gaifman and Vaughan Pratt. Partial order models of concurrency and the computation of functions. In *Proceedings; Symposium on Logic in Computer Science*, pages 72–85, Ithaca, New York, June 1987. Computer Society of the IEEE.

[21] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[22] Jay Loren Gischer. *Partial Orders and the Axiomatic Theory of Shuffle*. PhD thesis, Stanford University, December 1984. Available as Computer Science Technical Report number STAN-CS-84-1033.

[23] Robert Goldblatt. *Topoi: The Categorial Analysis of Logic*. Studies in Logic and the Foundations of Mathematics Vol 98. North-Holland, revised edition, 1983.

[24] Jan Grabowski. On partial languages. *Fundamenta Informaticae*, IV(2):427–498, 1981.

[25] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.

[26] Peter T. Johnstone. *Stone Spaces*. Cambridge Studies in Advanced Mathematics 3. Cambridge University Press, 1982.

[27] Gilles Kahn. The semantics of a simple language for parallel programming. In *Information Processing 74*, pages 471–475. North-Holland, 1974.

[28] G. M. Kelly and Ross Street. Review of the elements of 2-categories. In Gregory M. Kelly, editor, *Category Seminar, Sydney 1972/73*, Lecture Notes in Mathematics 420, pages 75–103. Springer-Verlag, 1974.

[29] G.M. Kelly. *Basic Concepts of Enriched Category Theory*. London Math. Soc. Lecture Notes 64. Cambridge University Press, 1982.

[30] F. William Lawvere. Metric spaces, generalized logic, and closed categories. *Rendiconti del Seminario Matematico e Fisico di Milano*, pages 135–166, 1973.

[31] Saunders Mac Lane. *Categories for the Working Mathematician*. Graduate Texts in Mathematics 5. Springer-Verlag, New York, 1971.

[32] Michael G. Main and David B. Benson. Functional behavior of nondeterministic and concurrent programs. *Information and Control*, 62:144–189, 1984.

[33] Zohar Manna and Amir Pnueli. Verification of concurrent programs: A temporal proof system. In J. W. de Bakker and J. van Leeuwen, editors, *Foundations of Computer Science: Distributed Systems*, pages 163–255. Mathematisch Centrum, Amsterdam, 1983.

[34] Narciso Martì-Oliet and José Meseguer. From Petri nets to linear logic. Technical Report SRI-CSL-89-4, SRI International Computer Science Laboratory, March 1989.

[35] José Meseguer and Ugo Montanari. Petri nets are monoids: A new algebraic foundation for net theory. In *Third Annual Symposium on Logic in Computer Science*, pages 155–164, Edinburgh, New York, July 1988. The Computer Society's Technical Committee on Mathematical Foundations of Computing, Computer Society Press, Washington DC.

[36] Robin Milner. *A Calculus of Communicating Systems.* Lecture Notes in Computer Science 92. Springer-Verlag, 1980.

[37] Robin Milner. Lectures on a calculus of communicating systems. In Broy [12], pages 205–228.

[38] U. Montanari and C. Simonelli. On distinguishing concurrency from nondeterminism. Technical Report 7, Progetto Finalizzato Informatica, Consiglio Nazionale delle Ricerche; Progetto P_1 Cnet, Pisa, Italy, October 1980.

[39] Mogens Nielsen, Gordon Plotkin, and Glynn Winskel. Petri nets, event structures and domains, part I. *Theoretical Computer Science*, 13:85–108, 1981.

[40] V. R. Pratt. On the composition of processes. In *Proceedings of the Ninth Annual ACM Symposium on the Principles of Programming Languages*, pages 213–223, January 1982.

[41] V. R. Pratt. Some constructions for order-theoretic models of concurrency. In *Proc. Conf. on Logics of Programs*, Lecture Notes in Computer Science 193, Brooklyn, 1985. Springer-Verlag.

[42] Vaughan Pratt. Modeling concurrency with partial orders. *International Journal of Parallel Processing*, 15(1):33–71, February 1986.

[43] Vaughan Pratt. Enriched categories and the Floyd-Warshall connection. In *First International Conference on Algebraic Methodology and Software Technology, AMAST*, pages 177–180, Iowa City, Iowa, May 1989.

[44] Wolfgang Reisig. *Petri Nets.* EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1985. (English translation of *Petrinetze*, Springer-Verlag, 1982.).

[45] P. Robert and J. Ferland. Généralisation de l'algorithme de Warshall. *Revue francaise d'Informatique et de Recherche Operationnelle*, 2(7):13–25, 1968.

[46] B. Roy. Transitivité et connexité. *Comptes Rendus Acad. Sci.*, 249:216–218, 1959.

[47] Andrzej Tarlecki. Bits and pieces of the theory of institutions. In David Pitt, Samson Abramsky, Axel Poigné, and David Rydeheard, editors, *Category Theory and Computer Programming*, Lecture Notes in Computer Science 240, pages 334–363. Springer-Verlag, Guildford, UK, 1985.

[48] M. Wand. Fixed-point constructions in order-enriched categories. *Theoretical Computer Science*, 8(1):13–30, 1979.

[49] S. Warshall. A theorem on Boolean matrices. *Journal of the ACM*, 9(1):11–12, 1962.

[50] Glynn Winskel. *Events in Computation.* PhD thesis, University of Edinburgh, December 1980. (Available as Computer Science department technical report CST-10-80).