Math. Struct. in Comp. Science (1993), *vol.* 3, *pp.* 259-276

Some Monoidal Closed Categories of Stable Domains and Event Structures

Guo-Qiang Zhang: *
Artificial Intelligence Laboratory
Department of EECS,
The University of Michigan
Ann Arbor, MI 48109

Received 3 January 1992; revised 26 June 1992

This paper introduces the following new constructions on stable domains and event structures: the tensor product, the linear function space, and the exponential. It gives rise to a monoidal closed category of dI-domains as well as one of stable event structures which can be used to interpret intuitionistic linear logic. Finally, the usefulness of the category of stable event structures for modeling concurrency and its relation to other models are discussed.

CONTENTS

1. Introduction

In [GL86], Girard and Lafont introduced intuitionistic linear logic with two (equivalent) formalisms: the sequent calculus and the combinators. In terms of category theory, as suggested in their paper, the combinator formulation corresponds to a linear category, *i.e.*, a symmetric monoidal closed category with finite products and coproducts [GL86,

: * On leave from Department of Computer Science, The University of Georgia.

MM89]. One of the well-known monoidal closed categories is the one of coherent spaces (with linear maps) [Gi86]. In fact, it is in this category that Girard discovered linear logic [Gi87], based on the crucial observation that the stable function space can be decomposed into more elementary linear constructions.

Coherent spaces † can be considered as special kinds of stable event structures [Wi87] with a trivial enabling relation. In this paper, we present monoidal closed categories of stable domains and stable event structures. The following new constructions on stable domains and event structures are introduced: tensor product, linear function space, and exponential. Other conventional constructions like product and sum also exist (see [Wi87]), hence the monoidal closed categories can be used to interpret intuitionistic linear logic in the sense of [Gi87, GL86] (see also [Se89]).

Stable event structures are closely related to the more abstract dI-domains (also called stable domains) on the one hand, and to the more concrete structure of Petri nets on the other [NPW81]. By showing that most of the constructions on coherent spaces generalize to stable event structures one hopes to open a new angle for a better understanding of the connection between linear logic and concurrency.

This paper owns to the work of Thierry Coquand [Co87] and Glynn Winskel [Wi88a]. I am grateful for Glynn Winskel for pointing to [GL86].

2. Stable Domains

The purpose of this section is to give a brief presentation of a monoidal closed category of stable domains (also called dI-domains). More details about the terminology used in this paper on domain theory can be found in [Zh91].

Definition 2.1. (Berry) A stable domain is an ω-algebraic, consistently complete cpo D which satisfies

Axiom d: $\forall x,\, y,\, z \in D.\; y \uparrow z \Longrightarrow x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z)$, and
Axiom I: $\forall d \in D^0.\;\; \mid \{\, x \mid x \sqsubseteq d \,\} \mid < \infty$.

Here D^0 is the set of finite elements of D, and $y \uparrow z$ means that there is an element above both y and z. In general, if X is bounded (above), we write $X\uparrow$. Axiom d expresses the distributive property and Axiom I says that in D every finite element dominates only a finite number of elements.

The morphisms of the category of stable domains are linear, stable functions.

Definition 2.2. A function f from a stable domain D to a stable domain E is stable if it is Scott-continuous, and it preserves meets of pairs of compatible elements, i.e.,

$$\forall x,\, y \in D.\;\; x \uparrow y \Longrightarrow f(x \sqcap y) = f(x) \sqcap f(y).$$

It is linear if moreover it preserves joins of bounded subsets of D, i.e.,

$$X \uparrow \Longrightarrow f(\bigsqcup X) = \bigsqcup fX.$$

: † Sometimes also called coherence spaces, but 'coherent spaces' is easier to pronounce, and is a term Girard himself uses in [Gi87].

Let f, g be stable functions from D to E. f is stably less than g, written $f \sqsubseteq_s g$, if

$$\forall x,\ y \in D.\ \ x \sqsubseteq y \Longrightarrow f(x) = f(y) \sqcap g(x).$$

The stable order provided above is sometimes also called the *Berry order*. For convenience, we write $[D \to_l E]$ for the cpo of linear, stable functions from D to E, and $[D \to_s E]$ for the cpo of stable functions from D to E (both under the stable order). It is well-known that $[D \to_l E]$ and $[D \to_s E]$ are both stable domains [Be78], [Zh92].

In dealing with stable domains and stable functions, the notion of a complete prime is useful. We say that p is a complete prime in a cpo D if for any bounded subset X of D, whenever $p \sqsubseteq \bigsqcup X$, there exists an $a \in X$ such that $p \sqsubseteq a$. We write D^p for the collection of complete primes of D. Of course $D^p \subseteq D^0$. For stable domains D^p is the set of join-irreducible compact elements.

Stable functions have the property that their values are totally determined by those at some minimal points. To describe the minimal points, we introduce the following definition.

Definition 2.3. Let $f : D \to E$ be a stable function, where D, E are stable domains. Define $\mu f \subseteq D^0 \times E^p$ to be the set of pairs such that $(a,\ b\,) \in \mu f$ if

$$f(a) \sqsupseteq b\ \&\ [\forall a' \sqsubseteq a.\ f(a') \sqsupseteq b \Longrightarrow a = a'].$$

The following is the key representation result that will be used in the proofs to follow.

Theorem 2.1. Let f, g be stable functions from D to E. $f \sqsubseteq_s g$ if and only if $\mu f \subseteq \mu g$. For $\{\ (\,a_i,\ b_i\,) \mid i \in I\ \} \subseteq D^0 \times E^p$,

$$\{\ (\,a_i,\ b_i\,) \mid i \in I\ \} = \mu f$$

for some $f \in [D \to_s E]$ if and only if

- $\forall J \subseteq^{fin} I.\,\{\,a_i \mid i \in J\,\}\!\uparrow \Longrightarrow \{\,b_i \mid i \in J\,\}\!\uparrow$,
- $a_i \uparrow a_j\ \&\ (b_i = b_j) \implies (a_i = a_j)$,
- $\forall b \in E^p.\ b_i \sqsupseteq b \implies \exists j.\ b_j = b\ \&\ a_i \sqsupseteq a_j$.

Moreover, f is linear, stable if and only if $\pi_0(\mu f) \subseteq D^p$, where π_0 is the projection to the first components.

Proof. The details of the proof is similar to that of Theorem 3.1 in [Zh92]. Note for any set $\{\,(a_i,\ b_i) \mid i \in I\,\} \subseteq D^0 \times E^p$ with the three properties mentioned in the theorem, the stable function f such that

$$\{\ (\,a_i,\ b_i\,) \mid i \in I\ \} = \mu f$$

can be obtained as the pointwise l.u.b. $\bigsqcup_{i \in I}[\,a_i, b_i\,]$, where

$$[\,a, b\,](x) = \begin{cases} b & \text{if } x \sqsupseteq a\ , \\ \perp & \text{otherwise.} \end{cases}$$

The monoidal closedness of stable domains involves a construction called the *tensor product.*

Definition 2.4. Let D, E be stable domains. The tensor product of D and E is the domain $D \otimes E$ with elements subsets $t \subseteq D^p \times E^p$ under set inclusion, where the t's are downwards closed sets such that both $\pi_0 t$ and $\pi_1 t$ are bounded.

$D \otimes E$ is a stable domain because it satisfies Axiom I and is prime-algebraic (see [Wi87], [Zh92]). This is a quick way to check that a domain is stable. It follows from the definition that the partial order of complete primes of $D \otimes E$ is isomorphic to $D^p \times E^p$, under the coordinatewise order.

Theorem 2.2. (Monoidal Closedness) For stable domains D, E and F, we have

$$(D \otimes E) \to_l F \cong D \to_l [E \to_l F].$$

Proof. The proof needs to use the fact that complete primes in the linear function space are functions f such that there is a pair $(p, q) \in \mu f$ with the property

$$(p', q') \in \mu f \Rightarrow p' \sqsubseteq [p \ \& \ q' \sqsubseteq q].$$

This is true because for any $(p', q') \in \mu f$, the set

$$\lceil (p', q') \rceil_f := \{(a, b) \mid (a, b) \in \mu f \ \& \ a \sqsubseteq p' \ \& \ b \sqsubseteq q'\}$$

determines a linear, stable function. Therefore, f is dominated by the least upper bound of the set

$$\{\lceil (p', q') \rceil_f \mid (p', q') \in \mu f\}.$$

From the definition of a complete prime, we must have

$$\mu f \subseteq \lceil (p', q') \rceil_f$$

for some (p', q') in μf; but this can happen only if

$$\mu f = \lceil (p', q') \rceil_f.$$

Back to the main proof, we define a mapping γ from $D \otimes E \to_l F$ to $D \to_l [E \to_l F]$, as follows:

$$\mu f \longmapsto \{(p, [(q, r)]_p) \mid ((p, q), r) \in \mu f\},$$

where

$$[(q, r)]_p := \{(q', r') \mid q' \sqsubseteq q \ \& \ r' \sqsubseteq r \ \& \ \exists p' \sqsubseteq p. \ ((p', q'), r') \in \mu f\}.$$

First, we must show that each $[(q, r)]_p$ corresponds to a prime, linear, stable function. For this we show that it is a trace set, i.e., it has the three properties (compatibility, minimality, and completeness) mentioned in Theorem 2.1.

- Compatibility is trivial because there is a maximal element in $[(q, r)]_p$.
- Minimality: Let $(q_1, r'), (q_2, r')$ be members of $[(q, r)]_p$ such that $q_1 \uparrow q_2$. Then there are p_1, p_2, both below p, such that

$$((p_1, q_1), r'), ((p_2, q_2), r') \in \mu f.$$

Hence $q_1 = q_2$, by minimality of μf.

- Completeness: Suppose $r' \sqsubseteq r$, where r' is a complete prime in F. Then, by the completeness of μf, there is some (p', q') such that
$$((p', q'), r') \in \mu f.$$
Moreover, $p' \sqsubseteq p$, $q' \sqsubseteq q$. Therefore, $(q', r') \in [(q, r)]_p$ and $[(q, r)]_p$ is complete.

Next, we show that
$$\gamma(\mu f) := \{(p, [(q, r)]_p) \mid ((p, q), r) \in \mu f\}$$
itself is a trace set. In the rest of the addendum, whenever we check compatibility, we check the simpler case of binary compatibility, just for notational reasons. It is easy to see how the binary case generalizes to the general case.

- Compatibility : For pairs $(p, [(q, r)]_p), (p', [(q', r')]_{p'})$ in $\gamma(\mu f)$ such that $p \uparrow p'$, we have $((p, q), r), ((p', q'), r') \in \mu f$. If $q \uparrow q'$, then the compatibility of μf ensures $r \uparrow r'$. We can easily see that in this case $[(q, r)]_p \cup [(q', r')]_{p'}$ is a trace set. If, on the other hand, q and q' are not compatible, we still need to show that
$$[(q, r)]_p \cup [(q', r')]_{p'}$$
is a trace set, which is routine to check. Therefore, the compatibility of p and p' implies the compatibility of $[(q, r)]_p$ and $[(q', r')]_{p'}$.
- Minimality: Let $(p_1, [(q, r)]_{p_1}), (p_2, [(q, r)]_{p_2})$ both be members of $\gamma(\mu f)$, and $p_1 \uparrow p_2$. Then $((p_1, q), r), ((p_2, q), r)$ are members of μf; so $p_1 = p_2$.
- Completeness: Suppose h is a complete prime in $[E \to_l F]$ and
$$\mu h \subseteq [(q, r)]_p.$$
We know that μh is of the form $\lceil (q', r') \rceil_h$ for some q', r'. By the completeness of μf, $r' \sqsubseteq r$ implies that there is some $((a, b), r')$ in μf such that $a \sqsubseteq p$, $b \sqsubseteq q$. Of course b must be equal to q' because both (b, r') and (q', r') are in $[(q, r)]_p$, and $b \uparrow q'$. For $[(q', r')]_a$ and μh are clearly compatible and they have the same maximal element (q', r'), it must be the case that $[(q', r')]_a = \mu h$. This proves the existence of some $a \sqsubseteq p$ such that $(a, \mu h)$ belongs to $\gamma(\mu f)$, which was what completeness needed.

It is clear that γ is one to one; we explain why it is onto. Let g be a linear, stable function in $D \to_l [E \to_l F]$. It is enough to show that the set
$$\beta(g) := \{((p, q), r) \mid (p, \lceil (q, r) \rceil_s) \in \mu g\}$$
is a trace set and γ maps it back to μg. Here s' are varying stable functions in $[E \to_l F]$, determined by g.

- Compatibility: Suppose $((p, q), r), ((p', q'), r') \in \beta(g)$ are such that
$$p \uparrow p', \;\; q \uparrow q'.$$
By the compatibility of μg, we know that $\lceil (q, r) \rceil_s$ and $\lceil (q', r') \rceil_{s'}$ are compatible. Hence $r \uparrow r'$.
- Minimality: Let $((p, q), r), ((p', q'), r) \in \beta(g)$ be such that $p \uparrow p'$, $q \uparrow q'$. From the compatibility of $\lceil (q, r) \rceil_s$ and $\lceil (q', r) \rceil_{s'}$ we get $q = q'$. Hence
$$\lceil (q, r) \rceil_s = \lceil (q, r) \rceil_{s'},$$

which in turn implies that $p = p'$.

- Completeness: Suppose $((p, q), r) \in \beta(g)$ and r' is a complete prime below r. By definition, $(p, \lceil (q, r) \rceil_s) \in \mu g$ for some $s \in [E \to_l F]$. The completeness of $\lceil (q, r) \rceil_s$ ensures the existence of some $q' \sqsubseteq q$ such that

$$(q', r') \in \lceil (q, r) \rceil_s.$$

But $\lceil (q', r') \rceil_s$ is a complete prime below $\lceil (q, r) \rceil_s$. So for some $p' \sqsubseteq p$,

$$(p', \lceil (q', r') \rceil_s) \in \mu g,$$

which is enough for the completeness of $\beta(g)$.

It remains to show that $\gamma(\beta(g)) = \mu g$. For any $((p, q), r) \in \beta(g)$, we have

$$(p, [(q, r)]_p) \in \gamma(\beta(g)).$$

Also, for some s,

$$(p, \lceil (q, r) \rceil_s) \in \mu g.$$

We show that $[(q, r)]_p = \lceil (q, r) \rceil_s$.

- "$\supseteq$": Let (q', r') be a member of $\lceil (q, r) \rceil_s$. Then $\lceil (q', r') \rceil_s$ is a complete prime dominated by $\lceil (q, r) \rceil_s$. The completeness of μg implies that there is some $p' \sqsubseteq p$ such that $(p', \lceil (q', r') \rceil_s) \in \mu g$. Hence, (q', r') is also a member of $[(q, r)]_p$.
- "$\subseteq$": Let (q', r') be any member of $[(q, r)]_p$. From the definition of γ, we know that there must be some $p' \sqsubseteq p$ such that

$$((p', q'), r') \in \beta(g).$$

This means for some linear, stable function $s' \in [E \to_l F]$,

$$(p', \lceil (q', r') \rceil_{s'}) \in \mu g.$$

By the completeness of $\lceil (q, r) \rceil_s$, there is some $q'' \sqsubseteq q'$ such that

$$(q'', r') \in \lceil (q, r) \rceil_s.$$

Also, the completeness of μg implies the existence of some $p'' \sqsubseteq p$ such that

$$(p'', \lceil (q'', r') \rceil_s) \in \mu g.$$

Accordingly, $((p'', q''), r')$ is also a member of $\beta(g)$. However, (p', q') and (p'', q'') are compatible. By the minimality of $\beta(g)$,

$$(p', q') = (p'', q''),$$

which implies that (q', r') is a member of $\lceil (q, r) \rceil_s$.

All the above seems natural. However, the construction of exponential is a bit of a surprise, since it is very different from that given by Girard for coherent spaces. Our treatment follows the guideline that the exponential should be used to decompose stable functions into linear ones.

The representation theorem (Theorem 2.1) indicates that stable functions and linear functions differ only in that the former are built from tokens (a, b) with $a \in D^0$, but the latter are built from (a, b) with a complete primes in D. Therefore, to decompose the stable functions into linear ones the exponential should be a construction ! such that for any stable domain D, $!D$ is a stable domain such that a is a finite element in D if and only if $!a$ is a complete prime in $!D$, and $a \sqsubseteq b$ if and only if $!a \sqsubseteq !b$. This analysis leads to the following definition.

Definition 2.5. Let D be a stable domain. The exponential of D is the domain $!D$ consisting of consistent, downwards closed subsets of D, ordered under inclusion.

$!D$ is again a stable domain due to the remark immediately after Definition 2.4. Notice that the complete primes of $!D$ are of the form

$$\lceil x \rceil = \{y \mid y \in D \;\&\; y \subseteq x\},$$

with x a finite element. One can see that there is a $1-1$, order preserving correspondence between the finite elements of D and the complete primes of $!D$. This is given by

$$x \longmapsto \lceil x \rceil.$$

The following theorem indicates that the exponential has the desired effect: it can be used to decompose a function space into a linear one.

Theorem 2.3. Let D, E be stable domains. Then

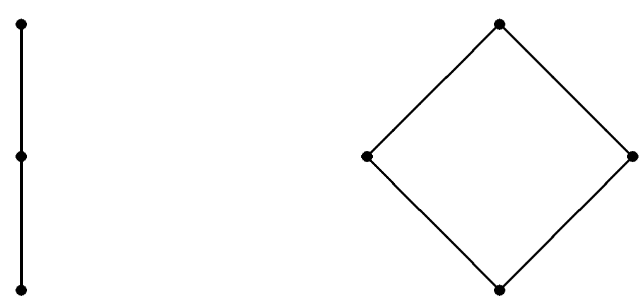
$$[D \to_s E] \cong (!D) \to_l E.$$

Proof. Again we use Theorem 2.1. The function β given by

$$\beta : (x, b) \longmapsto (\lceil x \rceil, b)$$

exhibits a 1-1 correspondence between $[D \to_s E]$ and $(!D) \to_l E$, which preserves the three conditions mentioned in Theorem 2.1.

The exponential given here, however, does not seem to be a direct generalization of the one given by Girard [Gi87]. Consider the exponential of the coherent space (see Section 4 for the definitions) with a single token. Below, the picture on the right is the result of applying Girard's exponential, and the one on the left is ours — It is simply not a coherent space!



3. Stable Event Structures

If we were only concerned with a domain theoretic model for intuitionistic linear logic, there would be less interest in developing a parallel framework of stable event structures. This is because dI-domains and stable event structures are equivalent categories. As indicated in the introduction, however, one of the purposes of the paper is to open a way to relate models of linear logic to those of concurrency. Models for concurrency are usually very concrete, such as Petri nets and event structures. A concrete model for intuitionistic linear logic should make the connection easier to establish. This section gives a detailed treatment of a monoidal closed category of stable event structures.

3.1. Event Structures

Event structures ([NPW81], [Wi87], [Wi88]) are models for processes of concurrent computation. An event structure is a description of a set of events in terms of a consistency and an enabling relation. The consistency relation indicates whether some events can occur together or not, and the enabling relation specifies the condition when a particular event may occur with regards to the occurrence of other events.

Definition 3.1. An event structure is a triple

$$\underline{E} = (\, E,\ Con,\ \vdash\,)$$

where

- E is a countable set of events,
- Con is a non-empty subset of $\mathrm{Fin}\,(E)$, the finite subsets of E called the consistency predicate which satisfies
$$X \subseteq Y \;\&\; Y \in Con \Longrightarrow X \in Con,$$
- $\vdash\, \subseteq Con \times E$ is the enabling relation which satisfies
$$(\, X \vdash e \;\&\; X \subseteq Y \;\&\; Y \in Con\,) \Longrightarrow Y \vdash e.$$

When we have the enabling relation $X \vdash e$ in an event structure, we say that the event e is enabled by the set of events X. In other words, if all the events in X have occurred, e may (but not necessarily) occur. Note that, following this intuition, an enabling of the form $\{e\} \vdash e$ does not contribute any information to an event structure. Therefore, if he likes, one can explicitly require in Definition 3.1 the following:

$$\text{if } X \vdash e \;\text{ then there exists } Y \subseteq X \;\text{ with } e \notin Y \;\text{ and } Y \vdash e.$$

However, that is hardly essential to our treatment.

A configuration of an event structure is a set of events which is consistent and in which each event has an enabling by a subset of the configuration. That is, a configuration is a set of events which have occurred by certain stage in a process.

In this paper lower case letters such as x, y usually range over sets of events, especially configurations, and upper case letters such as X, Y range over *finite sets of events.*

Definition 3.2. Let $\underline{E} = (E, \mathit{Con}, \vdash)$ be an event structure. A configuration of $\underline{E}$ is a subset $x \subseteq E$ which is

- consistent: $\forall X \subseteq^{fin} x.\, X \in \mathit{Con}$,
- secured: $\forall e \in x\, \exists e_0, e_1, \cdots, e_n \in x.\, e_n = e$ & $\forall i \leq n.\, \{ e_k \mid 0 \leq k \leq i-1 \} \vdash e_i$.

We write the set of configurations of an event structure $\underline{E}$ as $\mathcal{F}(\underline{E})$.

There is a special class of event structures for which each configuration determines a partial order of causal dependency on the events. Intuitively, with respect to a configuration x, an event e_1 causally depends on an event e_0 if the occurrence of the event e_0 is necessary for the occurrence of e_1. Event structures of this kind are called *stable*.

Definition 3.3. An event structure $\underline{E}$ is stable if it satisfies the following axiom:

$$(X \vdash e \;\&\; Y \vdash e \;\&\; X \cup Y \cup \{e\} \in \mathit{Con}) \Longrightarrow X \cap Y \vdash e.$$

We mentioned that a stable event structure is an event structure each configuration of which determines a partial order of causal dependency on the events of the configuration. This partial order is specified as follows.

Definition 3.4. Let $\mathcal{F}(\underline{E})$ be the family of configurations of a stable event structure $\underline{E}$. Let $x \in \mathcal{F}(\underline{E})$ and $e, e' \in x$. Define

$$e' \leq_x e \text{ if } \forall y \in \mathcal{F}(\underline{E}).\, (e \in y \;\&\; y \subseteq x \Longrightarrow e' \in y),$$
$$\lceil e \rceil_x = \bigcap \{ y \in \mathcal{F}(\underline{E}) \mid e \in y \;\&\; y \subseteq x \}.$$

One can show that $\leq_x$ is a partial order and $\lceil e \rceil_x = \{ e' \in x \mid e' \leq_x e \}$.

Stable event structures are a representation of stable domains. A result of Winskel [Wi87] says that for any stable event structure $\underline{E}$, the set of configurations $\mathcal{F}(\underline{E})$ under inclusion is a stable domain, and for any stable domain D, one can get a stable event structure whose events are the complete primes of D. Given a stable event structure $\underline{E}$, the complete primes in $\mathcal{F}(\underline{E})$ are exactly those configurations of the form $\lceil e \rceil_x$, mentioned earlier. Therefore, this gives a more concrete description of complete primes, which is sometimes easier to use in the case of stable event structures.

3.2. Linear Maps

This subsection introduces the category **SEL**, stable event structures with linear maps.

First there is a useful notion of minimal entailment [Wi87]. Let $\underline{E} = (E, \mathit{Con}, \vdash)$ be a stable event structure. Write $X \vdash_{min} e$ if $X \vdash e$, and moreover, $Y \vdash e$ and $Y \subseteq X$ implies $Y = X$. Suppose $Y \vdash e$. Then for

$$X = \bigcap \{ X' \mid X' \subseteq Y \;\&\; X' \vdash e \}$$

we have $X \vdash_{min} e$. Suppose $X \vdash_{min} e$ and $Z \vdash_{min} e$. If $X \cup Z \cup \{e\} \in \mathit{Con}$, then $Z = X$. Of course interesting minimal entailments are those relations $X \vdash_{min} e$ with $e \notin X$.

Definition 3.5. Let $\underline{E} = (\,E,\ Con_{\underline{E}},\ \vdash_{\underline{E}}\,)$, $\underline{F} = (\,F,\ Con_{\underline{F}},\ \vdash_{\underline{F}}\,)$ be stable event structures. A linear map from $\underline{E}$ to $\underline{F}$ is a relation $R \subseteq \mathrm{P}_{\underline{E}} \times F$ which satisfies

1. Compatibility : $(\,\forall i \in I\,.\ x_i\, R\, e_i\,)\ \&\ \bigcup x_i \in Con_{\underline{E}} \Longrightarrow \{\, e_i \mid i \in I\,\} \in Con_{\underline{F}}$,
2. Minimality : $(\, x \cup y \in \mathcal{F}(\underline{E})\ \ \&\ x\, R\, e\ \&\ y\, R\, e\,) \Longrightarrow x = y$, and
3. Completeness: $x\, R\, e \ \Longrightarrow \exists X.(X \vdash_{min} e\ \&\ \forall e' \in X \exists x' \subseteq x.\ x'\, R\, e')$.

Here $\mathrm{P}_{\underline{E}}$ stands for the set of complete primes of $\mathcal{F}(\underline{E})$, and I is finite.

In some previous cases, people use some kind of stable functions on the cpo of configurations as morphisms in various categories of event structures. This is unsatisfactory because stable functions are morphisms for the category of dI-domains. Event structures should have their own morphisms, just as information systems have approximable mappings as morphisms [Sc82], instead of Scott continuous functions.

Our definition of linear maps is motivated from the characterization theorem on the structure of stable functions (Theorem 2.1). Although it looks slightly complicated, I doubt anything better could be done, given that stable functions should not be used directly as morphisms here.

From a concurrency prospective, a linear map from $\underline{E}$ to $\underline{F}$ corresponds to a *simulation* of events in $\underline{F}$ by *computations* in $\underline{E}$. Under this interpretation, the compatibility axiom says that compatible computations should simulate consistent events. The minimality axiom ensures that in any possible simulation history, different computations should never simulate the same event. This is important as far as synchronization is concerned. One could justifiably say that an event e in $\underline{F}$ is always synchronized with e itself. However, if e is simulated by distinct stages in a computation history, it is impossible for these different computation stages to 'synchronize'. Finally, the completeness axiom guarantees that whenever an event e is simulated by a computation x, the 'pre-histories' (subsets of x which are configurations) of x should simulate a minimal set of events necessary for e to occur. In other words, x should be a 'sound' simulation of e.

Stable event structures with linear maps form a category. The identity Id is given by $\lceil e \rceil_x\, Id\, e$ if and only if $e \in x$ and $x \in \mathcal{F}(E)$. To compose linear maps, define $\tilde{R}$, for a linear map R, to be a relation such that $a\, \tilde{R}\, b$ if and only if there exist $a_i,\ p_i$ such that $a_i\, R\, p_i$, $1 \le i \le n$, $a = \bigcup_{1 \le i \le n} a_i \in \mathcal{F}(\underline{E})$, and $b = \{\, p_i \mid 1 \le i \le n\,\} \in \mathcal{F}(\underline{F})$. It is not difficult to check that $\tilde{R}$ has the following properties:

1. $(\,\forall i \in I\,.\ a_i\, \tilde{R}\, b_i\,)\ \&\ \bigcup a_i \in Con_{\underline{E}} \Longrightarrow \bigcup b_i \in Con_{\underline{F}}$,
2. $(\, a \cup b \in \mathcal{F}(\underline{E})\ \ \&\ a\, \tilde{R}\, c\ \&\ b\, \tilde{R}\, c) \Longrightarrow a = b$, and
3. $a\, \tilde{R}\, b \supseteq c \Longrightarrow \exists a' \subseteq a.\ a'\, \tilde{R}\, c$.

Now we can compose linear mappings $R : \underline{E} \to \underline{F}$ and $S : \underline{F} \to \underline{G}$. Define $R \circ S$ to be a relation on $P_{\underline{E}} \times C$ by letting

$$a\,(R \circ S)\, p \Longleftrightarrow \exists b.\ a\, \tilde{R}\, b\ \&\ b\, S\, p.$$

From the properties of $\tilde{R}$ one can see that $R \circ S : \underline{E} \to \underline{G}$ is indeed a linear map.

The construction of linear function space is given as follows.

Definition 3.6. Let $\underline{E_0} = (E_0, Con_0, \vdash_0)$ and $\underline{E_1} = (E_1, Con_1, \vdash_1)$ be stable event structures. Their linear function space, $[\underline{E_0} \multimap \underline{E_1}]$, is defined as the structure $(E, Con, \vdash,)$ where

$$\begin{aligned}
&E = \{(\lceil e_0 \rceil_x, e_1) \mid e_0 \in x \in \mathcal{F}(\underline{E_0}), e_1 \in E_1\},\\
&X \in Con \text{ if and only if } \forall Y \subseteq X. \bigcup \pi_0 Y \in Con_0 \implies \pi_1 Y \in Con_1 \;\&\\
&\qquad\qquad \forall a, b \in X.(\pi_0 a \cup \pi_0 b \in \mathcal{F}(\underline{E_0}) \;\&\; \pi_1 a = \pi_1 b) \implies a = b,\\
&X \vdash (x, e_1) \text{ if and only if } \{e \mid (y, e) \in X \;\&\; y \subseteq x\} \vdash_1 e_1.
\end{aligned}$$

Note that $\lceil e \rceil_x$ is always finite.

That the construction of linear function space is indeed what we wanted (given the linear maps) is confirmed in the following theorem.

Theorem 3.1. The linear function space of two stable event structures is a stable event structure. R is a configuration in $\mathcal{F}[\underline{E_0} \multimap \underline{E_1}]$ if and only if $R: \underline{E_0} \to \underline{E_1}$ is a linear map.

Proof. The proof for the first part is routine.

To prove the second part, let R be a configuration in $\mathcal{F}[\underline{E_0} \multimap \underline{E_1}]$. It is clear that R is compatible and minimal. Assume $(x, e) \in R$. Then (x, e) must be secured. By the definition of the enabling in the linear function space, this means, in particular, that e is secured in $\pi_1 R'$, the projection of a subset of R into the second component. Furthermore, the first components of R' must be contained in x. This shows R is complete and hence a linear map.

By the definition of the enabling relation of the linear function space, a linear map is a configuration.

One can show further that there is a 1-1, order preserving correspondence between the linear maps and linear (stable) functions. That is why the linear maps are called 'linear'.

Theorem 3.2. There is a 1-1, order preserving correspondence between

$$\{ f \mid f: \mathcal{F}(\underline{E_0}) \to \mathcal{F}(\underline{E_1}) \text{ is linear, stable} \}$$

and

$$\{ R \mid R: \underline{E_0} \to \underline{E_1} \text{ is a linear map} \}.$$

Proof. The 1-1 correspondence is determined by a pair of functions. We give this pair in the following, but leave out the details of the proof.

Any linear, stable function $f: \mathcal{F}(\underline{E_0}) \to \mathcal{F}(\underline{E_1})$ gives rise to a linear map

$$\mu f = \{(x, e) \mid x \in P_{\underline{E_0}} \;\&\; e \in f(x) \;\&\; [(x' \subseteq x \;\&\; e \in f(x')) \implies x' = x]\}.$$

On the other hand, any linear map R induces a linear, stable function

$$Pt(R) = \lambda x.\{e \in E_1 \mid \exists y \subseteq x. (y, e) \in R\}.$$

3.3. Monoidal Closedness

We now introduce tensor products on stable event structures and show that **SEL** is monoidal closed.

Definition 3.7. Let $\underline{E_0} = (\, E_0, Con_0, \vdash_0 \,)$ and $\underline{E_1} = (\, E_1, Con_1, \vdash_1 \,)$ be stable event structures. Their tensor product, $\underline{E_0} \otimes \underline{E_1}$, is defined as the structure $(\, E,\, Con,\, \vdash,\,)$ where

$$\begin{aligned}
&E = E_0 \times E_1,\\
&X \in Con \iff \pi_0 X \in Con_0 \;\&\; \pi_1 X \in Con_1, \quad \text{and}\\
&X \vdash e \iff \exists X_0\,, X_1.\; X_0 \vdash_0 \pi_0 e \;\&\; X_1 \vdash_1 \pi_1 e \;\&\; X_0 \times X_1 \subseteq X.
\end{aligned}$$

It is interesting to note that the consistency predicate for tensor product is the same as the one generated by the set

$$\{X_0 \times X_1 \mid X_0 \in Con_0 \;\&\; X_1 \in Con_1\}.$$

Proposition 3.1. If $\underline{E_0}$ and $\underline{E_1}$ are stable then so is $\underline{E_0} \otimes \underline{E_1}$.

Proof. We check the stability axiom. Assume $X \vdash e$, $Y \vdash e$, and $X \cup Y \cup \{e\} \in Con$. By definition, there exist X_0, X_1, Y_0, and Y_1, such that

$$\begin{array}{ll}
X_0 \vdash_0 \pi_0 e, & X_1 \vdash_1 \pi_1 e,\\
Y_0 \vdash_0 \pi_0 e, & Y_1 \vdash_1 \pi_1 e,\\
X \supseteq X_0 \times X_1, & Y \supseteq Y_0 \times Y_1.
\end{array}$$

Also we have $X_0 \cup Y_0 \cup \{\pi_0 e\} \in Con_0$ and $X_1 \cup Y_1 \cup \{\pi_1 e\} \in Con_1$. Therefore $X_0 \cap Y_0 \vdash_0 \pi_0 e$ and $X_1 \cap Y_1 \vdash_1 \pi_1 e$, by the stability of $\underline{E_0}$ and $\underline{E_1}$. However, $X \cap Y \supseteq (X_0 \cap Y_0) \times (X_1 \cap Y_1)$. Hence $X \cup Y \vdash e$.

Let $x \in \mathcal{F}(\underline{E_0} \otimes \underline{E_1})$. We have $\pi_0 x \in \mathcal{F}(\underline{E_0})$. In fact, $\pi_0 x$ is clearly consistent and, if $e_0 \in \pi_0 x$, then there is some e_1 such that $(e_0,\, e_1) \in x$. The way enabling is defined for tensor product ensures that when $(e_0,\, e_1)$ is secured in x, e_0 is secured in $\pi_0 x$. Hence $\pi_0 x$ is a configuration of $\underline{E_0}$. Similarly $\pi_1 x \in \mathcal{F}(\underline{E_1})$. From this we know that for any $(e_0,\, e_1)$ in x, $\lceil e_0 \rceil_{\pi_0 x} \times \lceil e_1 \rceil_{\pi_1 x} \subseteq x$.

Suppose, on the other hand, that x is a subset of $E_0 \times E_1$ with the property that $\pi_0 x \in \mathcal{F}(\underline{E_0})$, $\pi_1 x \in \mathcal{F}(\underline{E_1})$, and

$$\forall (e_0,\, e_1) \in x \exists x_0 \in \mathcal{F}(\underline{E_0}),\; x_1 \in \mathcal{F}(\underline{E_1}).\, (e_0,\, e_1) \in x_0 \times x_1 \subseteq x.$$

x is obviously a consistent set. Also, since $e_0 \in x_0$ and $e_1 \in x_1$,

$$\lceil e_0 \rceil_{\pi_0 x} \times \lceil e_1 \rceil_{\pi_1 x} \subseteq x_0 \times x_1 \subseteq x.$$

Hence $(e_0,\, e_1)$ is secured in x. In summary, we have proved the following proposition.

Proposition 3.2. Let $\underline{E_0} = (\, E_0, Con_0, \vdash_0 \,)$ and $\underline{E_1} = (\, E_1, Con_1, \vdash_1 \,)$ be stable event structures. We have $x \in \mathcal{F}(\underline{E_0} \otimes \underline{E_1})$ if and only if

- $\pi_0 x \in \mathcal{F}(\underline{E_0}) \;\&\; \pi_1 x \in \mathcal{F}(\underline{E_1})$, and
- $\forall (e_0,\, e_1) \in x \exists x_0 \in \mathcal{F}(\underline{E_0}),\; x_1 \in \mathcal{F}(\underline{E_1}).\, (e_0,\, e_1) \in x_0 \times x_1 \subseteq x.$

We also have

Proposition 3.3. Let $\underline{E_0}$, $\underline{E_1}$ and $\underline{E_2}$ be stable event structures. Then

$$(\underline{E_0} \otimes \underline{E_1}) \otimes \underline{E_2} \cong \underline{E_0} \otimes (\underline{E_1} \otimes \underline{E_2}).$$

This statement is easy to justify. It indicates that the tensor product is associative up to an isomorphism.

Theorem 3.3. (Monoidal Closedness) Let $\underline{E_0}$, $\underline{E_1}$, and $\underline{E_2}$ be stable event structures. Then

$$(\underline{E_0} \otimes \underline{E_1}) \multimap \underline{E_2} \cong \underline{E_0} \multimap [\underline{E_1} \multimap \underline{E_2}].$$

The isomorphism $\underline{E} \cong \underline{E'}$ amounts to saying that there is a $1-1$ correspondence between E and E', which preserves the consistency and the entailment relations. A more categorical proof of monoidal closedness requires one to follow the definitions in [Mc71]. One can achieve such rigor but that is not the purpose here.

Proof. The isomorphism is given by

$$\theta:\ (\lceil(e_0,\, e_1)\rceil_x,\ e_2\,) \longmapsto (\lceil e_0\rceil_{\pi_0 x},\ (\lceil e_1\rceil_{\pi_1 x},\ e_2\,)\,).$$

This θ is well defined. Recall that whenever we write $\lceil e\rceil_x$, we require that $e \in x$. In addition, for any stable event structure, given any configuration x and any event e in x, $\lceil e\rceil_x$ is always a configuration. Proposition 3.2 indicates that when x is a configuration of the tensor product $\underline{E_0}\otimes\underline{E_1}$, $\pi_0 x$ and $\pi_1 x$ are configurations of $\underline{E_0}$ and $\underline{E_1}$, respectively. Therefore, $\lceil e_0\rceil_{\pi_0 x}$ and $\lceil e_1\rceil_{\pi_1 x}$ are uniquely specified (here we do not have to care about if e_0 or $\pi_0 x$ is unique or not).

Clearly θ is onto as $x_0 \times x_1$ is a configuration in the tensor product if x_0 and x_1 are configurations. It is then clear that θ is a bijection. Write

$$\begin{aligned}
(\underline{E_0} \otimes \underline{E_1}) \multimap \underline{E_2} &= (\,P,\ Con_P, \vdash_P\,),\\
\underline{E_0} \multimap [\underline{E_1} \multimap \underline{E_2}] &= (\,F,\ Con_F, \vdash_F\,),\\
[\underline{E_1} \multimap \underline{E_2}\,] &= (\,E,\ Con_E, \vdash_E\,).
\end{aligned}$$

Directly following the definitions one can check that $X \in Con_P$ if and only if $\theta X \in Con_F$, and $X \vdash_P e$ if and only if $\theta X \vdash_F \theta e$.

3.4. Exponential

There is a construction of exponential on stable event structures which can be used to decompose the stable function space into a linear function space. To fully explain the decomposition, let us recall Winskel's construction of the function space of stable event structures.

Definition 3.8. (Winskel) Let $\underline{E_0} = (\,E_0,\ Con_0,\ \vdash_0\,)$ and $\underline{E_1} = (E_1,\ Con_1,\ \vdash_1)$ be stable event structures. Their function space, $[\,\underline{E_0} \to \underline{E_1}]$, is defined to be the event structure $(\,E,\ Con,\ \vdash\,)$ with events E consisting of pairs $(\,x,\ e\,)$ where x is a finite configuration in $\mathcal{F}(\,\underline{E_0}\,)$ and $e \in E_1$, a consistency predicate Con given by

$\{\,(\,x_i,\ e_i\,) \mid 0 \le i \le n-1\,\} \in Con$ if and only if

- $\forall I \subseteq \{\,0,\ 1,\ \cdots,\ n-1\,\}.\ \bigcup_{i\in I} x_i \in Con_0 \Longrightarrow \{\,e_i \mid i \in I\,\} \in Con_1$
- $\forall i,\ j \le n.\ (\,x_i \cup x_j \in \mathcal{F}(\underline{E_0})\ \&\ e_i = e_j\,) \Longrightarrow x_i = x_j,$

and an enabling relation given by

$$\{\,(\,x_i,\ e_i\,) \mid 0 \le i \le n-1\,\} \vdash (\,x,\ e\,) \ \text{ if and only if } \{\,e_i \mid x_i \subseteq x\,\} \vdash_1 e.$$

Comparing the events in the function space with those in a linear function space we notice that the only difference is that in the former we use (x, e) with x finite configurations but in the latter we use $(\lceil e \rceil_x, e')$. Therefore all the finite configurations of $\underline{E}$ should correspond to the prime elements of $!\underline{E}$ such that the appropriate orders are preserved and reflected. Thus we have the following definition of exponential.

Definition 3.9. Let $\underline{E} = (E, Con, \vdash)$ be a stable event structure. The exponential of $\underline{E}$ is the event structure

$$!\underline{E} = (E', Con', \vdash')$$

where E' consists of the finite configurations of $\underline{E}$, $X \in Con'$ if and only if $\bigcup X \in E'$, and $\vdash'$ is a n enabling relation generated by $\{ x \} \vdash' y$ if $y = x \cup \{e\}$ for some e.

Clearly exponential preserves stability of event structures. There is a different definition of ! which gives the same configurations. The consistency relation is the same but the enabling relation is given by $X \vdash'' y$ if and only if X is consistent and

$$\{x \mid x \subset y \ \&\ x \in \mathcal{F}(\underline{E}) \} \subseteq X.$$

It does not matter which enabling relation is used since as long as it gives the same configurations, the decomposition in the following Theorem 3.4 remains sound.

One can easily check that the exponential of a stable event structure is a stable event structure. In fact, the exponential is not only stable, but also prime [Wi87]. The nice thing about it is that in a prime event structure, one can omit the subscript x in any of its configurations $\lceil e \rceil_x$.

From the fact that the complete primes of $\mathcal{F}(!\underline{E})$ are of the form

$$\lceil x \rceil = \{y \mid y \in \mathcal{F}(\underline{E}) \ \&\ y \subseteq x\},$$

one can see that there is a $1-1$, order preserving correspondence between the finite configurations of $\underline{E}$ and the complete primes of $!\underline{E}$, given by

$$x \longmapsto \lceil x \rceil.$$

Theorem 3.4. Let $\underline{E_0}$, $\underline{E_1}$ be stable event structures. Then

$$[\underline{E_0} \to \underline{E_1}] \cong (!\underline{E_0}) \multimap \underline{E_1}.$$

Proof. The function δ given by

$$\delta : (x, e) \longmapsto (\lceil x \rceil, e)$$

exhibits a 1-1 correspondence between $[\underline{E_0} \to \underline{E_1}]$ and $(!\underline{E_0}) \multimap \underline{E_1}$ which preserves the consistency and the entailment relations.

4. Concurrency

This section presents an attempt to relate the monoidal closed category of stable event structures to the category of event structures for modeling concurrency. We show that there is an adjunction between the two categories when the event structures are restricted to coherent spaces. We also indicate where the difficulty is in trying to generalize this result.

A coherent space is a special kind of stable event structure $(E, Con, \vdash)$ where *Con* is determined by a symmetric, irreflexive relation $\#$ (the conflict relation), and $\vdash$ is trivial in the sense that for every event $e \in E$, we have $\emptyset \vdash e$. Thus we say a set of events X consistent if $\forall e, e' \in X.\ \neg(e \# e')$. For technical reasons, here we allow E to be any set. It is convenient to write a coherent space in the form $(E, \#)$ since the enabling relation is trivial.

It is well-known that coherent spaces with linear maps form a monoidal closed category. The linear maps are exactly the same as those on stable event structures (but restricted to coherent spaces).

Definition 4.1. Let $\underline{E_0} = (E_0, \#_0)$, $\underline{E_1} = (E_1, \#_1)$ be coherent spaces. A linear map from $\underline{E_0}$ to $\underline{E_1}$ is a relation $R \subseteq E_0 \times E_1$ which satisfies

$$\begin{array}{l} 1.\ (e_0 R e_1\ \&\ e_0' R e_1'\ \&\ e_1 \#_1 e_1') \Longrightarrow e_0 \#_0 e_0', \\ 2.\ [e_0 R e\ \&\ e_0' R e\ \&\ \neg(e_0 \#_0 e_0')] \Longrightarrow e_0 = e_0'. \end{array}$$

Note that the completeness axiom as required for a linear map (see Definition 3.5) is vacuously true.

The category of event structures which gives non-interleaving semantics to concurrent languages like CCS has a different kind of morphisms: the *partially synchronous morphisms* [Wi87], [Wi88]. There are constructions of product and coproduct in this category, which are used to model parallel composition and nondeterministic choices of CCS like processes. Originally the objects of this category are stable event structures. When restricted to coherent spaces, partially synchronous morphisms can be simplified as follows.

Definition 4.2. Let $\underline{E_0} = (E_0, \#_0)$, $\underline{E_1} = (E_1, \#_1)$ be coherent spaces. A partially synchronous morphism from $\underline{E_0}$ to $\underline{E_1}$ is a partial function $\theta : E_0 \rightarrow E_1$ which satisfies

$$\begin{array}{l} 1.\ \theta(e) \#_1 \theta(e') \Longrightarrow e \#_0 e', \\ 2.\ [\theta(e) = \theta(e')\ \&\ \neg(e \#_0 e')] \Longrightarrow e = e'. \end{array}$$

Note that by convention when we write $\theta(e) \#_1 \theta(e')$ or $\theta(e) = \theta(e')$, $\theta(e)$ and $\theta(e')$ are both defined.

Comparing the definition of a linear map to that of a partially synchronous morphism, one can see that partially synchronous morphisms are special kind of linear maps. The only difference between these two notions is that a linear map is determined by a relation, while a partially synchronous morphism is determined by a partial function.

The reason for using a (partial) function for a partially synchronous morphism may be traced back to the synchronization mechanism for CCS processes. The synchronization of two CCS processes is achieved by communication through handshake, an indivisible action in which a message is simultaneously emitted by one process and received by the other [Mi89]. Handshakes happen between two parties only. CCS does not allow a party to handshake with more than one party at a given time. Therefore, the 'communication partner of' a process determines a partial function.

From a concurrency point of view, a linear map supports the synchronization of more than two events. Two pairs of events (e, e_1), (e, e_2) in the relation R can be understood as the synchronization of the event e with e_1 and e_2. There are many practical examples of synchronization among more than two parties. A conference call enabled by a telephone company is one such example.

However, linear maps seem to allow more than that. Potentially, one event can synchronize with infinitely many other events. Consider the following sequence of relations between a coherent space with a single event e and one with the event set ω and the empty conflict relation:

$$\begin{array}{l} R_0 = \emptyset, \\ R_1 = \{(e, 0)\}, \\ \ldots, \\ R_i = \{(e, j) \mid j < i\}, \\ \ldots \end{array}$$

Each R_i corresponds to a linear map, and the subset relation on them specifies the stable order. Therefore, the least upper bound (in the linear function space) of the sequence R_i is $\{(e, j) \mid j \in \omega\}$.

Our purpose is to establish an adjunction between the category of coherent spaces with partial synchronous morphisms (written as **CPS**) and the category of coherent spaces with linear maps (written as **CLI**). The foregoing discussions suggest that we should find a construction $\natural$ on coherent spaces $\underline{E_1}$ such that

$$\underline{E_0} \to_{lin} \underline{E_1} \cong \underline{E_0} \to_{ps} (\natural \underline{E_1}),$$

where $\underline{E_0} \to_{lin} \underline{E_1}$ stands for the set of linear maps from $\underline{E_0}$ to $\underline{E_1}$, and $\underline{E_0} \to_{ps} (\natural \underline{E_1})$ stands for the set of partial synchronous morphisms form $\underline{E_0}$ to $\natural \underline{E_1}$. With respect to the example mentioned in the previous paragraph, the linear map $\{(e, j) \mid j \in \omega\}$ should correspond to some kind of partial synchronous morphism $\{(e, \omega)\}$. This implies that the events of $\natural \underline{E_1}$ should be non-empty, consistent subsets of E_1 – not merely finite consistent subsets. This is exactly the technical reason that takes us out of countable event sets.

Definition 4.3. Let $\underline{E} = (E, \#)$ be a coherent space. Define $\natural \underline{E}$ to be the structure $(E', \#')$, where

$$\begin{array}{l} E' = \{x \mid x \subseteq E \ \& \ x \neq \emptyset \ \& \ \forall e, e' \in x. \neg(e \# e')\}, \\ x \#' y \Longleftrightarrow [(x \cap y \neq \emptyset \ \& \ x \neq y) \ \text{ or } \ \exists e \in x \exists e' \in y. e \# e']. \end{array}$$

This definition defines a coherent space because $\#'$ is a symmetric, irreflexive relation on E'. Now we are in a position to prove the following theorem.

Theorem 4.1. For coherent spaces $\underline{E_0} = (E_0, \#_0)$ and $\underline{E_1} = (E_1, \#_1)$, we have

$$\underline{E_0} \to_{lin} \underline{E_1} \cong \underline{E_0} \to_{ps} (\natural \underline{E_1}).$$

Proof. For any linear map $R : \underline{E_0} \to \underline{E_1}$, let

$$f(R) : E_0 \to \{x \mid x \subseteq E \ \& \ x \neq \emptyset \ \& \ \forall e, e' \in x. \neg(e \# e')\}$$

be a partial function such that $f(R)(e)$ is defined to be $\{e' \mid e' \in E_1 \ \& \ eRe'\}$ if it is not empty. Clearly $f(R)(e)$ is consistent. We show that $f(R)$ is a partially synchronous morphism from $\underline{E_0}$ to $\natural\underline{E_1}$. Assume $f(R)(e)\#f(R)(e')$. By Definition 4.3, either there exist $a \in f(R)(e)$ and $b \in f(R)(e')$ such that $a\#_1 b$, or $f(R)(e)$ and $f(R)(e')$ are different sets with a non-empty intersection. For the first case we have $e\#_0 e'$, by the properties of R. For the second case, there must be an $a \in E_1$ such that eRa and $e'Ra$. Again by the properties of R, we must have $e\#_0 e'$ since $f(R)(e) \neq f(R)(e')$ implies $e \neq e'$. We also have

$$[f(R)(e) = f(R)(e') \;\&\; \neg(e\#_0 e')] \Longrightarrow e = e'$$

since $f(R)(e) = f(R)(e')$ implies the non-emptyness of $f(R)(e)$. Therefore, $f(R)$ is indeed a partially synchronous morphism.

On the other hand, for any partially synchronous morphism

$$\theta : \underline{E_0} \to \natural\underline{E_1},$$

let

$$g(\theta) \subseteq E_0 \multimap E_1$$

be the relation such that $(e, e') \in g(\theta)$ if and only if $e' \in \theta(e)$. We show that $g(\theta)$ is a linear map from $\underline{E_0}$ to $\underline{E_1}$. Assume $(e_0, e_1), (e_0', e_1') \in g(\theta)$ and $e_1\#_1 e_1'$. By Definition 4.3, $\theta(e_0)\#\theta(e_0')$. Since θ is a partially synchronous morphism, we have $e_0\#_0 e_0'$. We now check that

$$[(e_0, e), (e_0', e) \in g(\theta) \;\&\; \neg(e_0\#_1 e_0')] \Longrightarrow e_0 = e_0'.$$

This is because the given conditions imply $\theta(e_0) \cap \theta(e_0') \neq \emptyset$ and $\neg(\theta(e_0)\#\theta(e_0'))$. This is only possible when $\theta(e_0) = \theta(e_0')$, which in turn implies $e_0 = e_0'$.

It is now a relatively easy task to check that f and g indeed give rise to the desired isomorphism.

The consequence of this result, put in categorical terms, is that the forgetful functor $U : \mathbf{CPS} \to \mathbf{CLI}$ has a right adjoint $F : \mathbf{CLI} \to \mathbf{CPS}$ as specified below.

Theorem 4.2. The forgetful functor

$$U : \mathbf{CPS} \to \mathbf{CLI}$$

has a right adjoint

$$F : \mathbf{CLI} \to \mathbf{CPS}$$

where $F(\underline{E}) = \natural\underline{E}$, and

$$F(R)(x) = \{e' \mid \exists e \in x.(e, e') \in R\}$$

for a linear map $R : \underline{E_0} \multimap \underline{E_1}$.

Proof. This follows from the previous theorem and the fact that

$$F(R) : F(\underline{E_0}) \to F(\underline{E_1})$$

is a partially synchronous morphism for any linear map $R : \underline{E_0} \multimap \underline{E_1}$.

Since right adjoints preserve limits such as products, categories **CLI** and **CPS** have the same product for modeling parallel composition of CCS processes.

The obvious next step is to extend the adjunction to stable event structures. To make things simpler, we work on prime event structures first. Recall that a prime event structure is a triple $\underline{E} = (E, Con, \leq)$, where E is a set of events, partially ordered by $\leq$, and Con is a consistency predicate which is downwards closed with respect to $\leq$ (the causal dependency order):

$$(X \in Con \;\&\; e \leq e' \in X) \Longrightarrow X \cup \{e\} \in Con.$$

Moreover, for any $e \in E$, the set $\lceil\lceil e \rceil\rceil = \{e' \mid e' \leq e \;\&\; e \neq e'\}$ is finite.

Our work in Section 3 can be easily extended to prime event structures [Wi87]. There is a monoidal closed category with the following linear maps as morphisms.

Definition 4.4. Let $\underline{E} = (\, E,\ Con_{\underline{E}},\ \vdash_{\underline{E}}\,)$, $\underline{F} = (\, F,\ Con_{\underline{F}},\ \vdash_{\underline{F}}\,)$ be prime event structures. A linear map from $\underline{E}$ to $\underline{F}$ is a relation $R \subseteq E \times F$ which satisfies

1. Compatibility : $(\, \forall i \in I \,.\, e_i \, R \, e'_i \,) \;\&\; \{e_i \mid i \in I\} \in Con_{\underline{E}} \Longrightarrow \{\, e'_i \mid i \in I \,\} \in Con_{\underline{F}}$,
2. Minimality : $(\, \{e_0, e_1\} \in Con_{\underline{E}} \;\&\; e_0 \, R \, e \;\&\; e_1 \, R \, e \,) \Longrightarrow e_0 = e_1$, and
3. Completeness: $e \, R \, e' \;\Longrightarrow \forall e'_1 \in \lceil\lceil e' \rceil\rceil \exists e'_0 \leq e .\, e'_0 \, R \, e'_1$.

Here I is a finite index set.

The definition of partially synchronous morphisms is almost the same on prime event structures as on stable event structures, if one recovers the enabling relation by letting

$$X \vdash e \;\text{ if and only if }\; \lceil\lceil e \rceil\rceil \subseteq X.$$

Definition 4.5. Let $\underline{E}_1 = (\, E_1,\ Con_1,\ \leq_1\,)$ and $\underline{E}_2 = (E_2,\ Con_2,\ \leq_2)$ be prime event structures. A partially synchronous morphism from $\underline{E}_1$ to $\underline{E}_2$ is a partial function $\theta : E_1 \to E_2$ on on events which satisfies

- $X \in Con_1 \Longrightarrow \theta X \in Con_2$,
- $(\{\, e,\ e' \,\} \in Con_1 \;\&\; \theta(e) = \theta(e')) \Longrightarrow e = e'$, and
- $\theta(e)$ is defined $\Longrightarrow \lceil\lceil \theta(e) \rceil\rceil \subseteq \theta \lceil\lceil e \rceil\rceil$.

To get an adjunction between the category of prime event structures with linear maps and the category of prime event structures with partially synchronous morphisms, a construction $\natural$ similar to the one on coherent spaces seems necessary. For a prime event structure $\underline{E}$, the events of $\natural\underline{E}$ should be non-empty subsets of E. The definition of the consistency relation on $\natural\underline{E}$ is similar to the one for coherent spaces. However, it is unclear to me at this moment how to get the appropriate definition of a causal dependency order on subsets of E so that a similar isomorphism as the one in Theorem 4.1 exists. None of the partial orders for powerdomains seems to work here, and the difficulty seems to be in making sure $\natural\underline{E}$ is a prime event structure. It could well be that everything works fine on stable event structures. However, we have to leave that as a research topic.

5. Discussion

We have introduced a monoidal closed category of dI-domains and a monoidal closed category of stable event structures. The exponential construction for decomposing stable functions given in this paper is not a generalization of the one given by Girard [Gi87]. To relate to models for concurrency, an adjunction between the category of coherent spaces with linear maps and the category of coherent spaces with partially synchronous morphisms is discovered. Such an adjunction indicates a way to break linear functions into even more basic constructions related to models of concurrency.

It seems an important characteristic of dI-domains which make them monoidal closed is that they are prime generated. The finitary axiom (Axiom I) does not seem to be an indispensable condition. This confirms the suggestion by Coquand [Co87] that prime generated domains are monoidal closed (either with linear, stable functions under Berry order, or with the linear, continuous functions under the extensional order).

Along the line of non-stable morphisms, a particularly simple exposition of prime algebraic domains in terms of prime event structures is worth brief mentioning. Here one can use a slightly different notion of prime event structures (than in the sense of [Wi87]). A prime event structure here is a triple $(E, \leq, Con)$ where E is a countable set of events, $\leq$ is a preorder, representing the causal relation between events, and Con is a consistency predicate as used for event structures. The ideal elements $\mathcal{F}(\underline{E})$ of a prime event structure are consistent, downwards (with respect to $\leq$) closed (or, left closed) subsets of E. The difference with the original definition [Wi87] is mainly that Axiom I is not imposed on $\leq$.

A domain D is prime (ω) algebraic iff it can be representated by a prime event structure. It is clear that prime event structures give rise to prime algebraic domains. On the other hand, given any prime algebraic domain $(D, \sqsubseteq)$, one can associate it with the prime event structure $(E^p, \sqsubseteq, Con)$, where a finite set of complete primes belows to Con iff it is bounded above. One can show that the set of configurations of this prime event structure is isomorphic to the original domain D.

Tensor product and linear function space can be introduced on prime event structures as follows. Given two prime event structures $(E_0, \leq_0, Con_0)$ and $(E_1, \leq_1, Con_1)$, Their tensor product $\underline{E_0} \otimes \underline{E_1}$ is the prime event structure $(E_0 \times E_1, \leq, Con)$ where $\leq$ and Con are both determined componentwise. Their linear space $[\underline{E_0} \multimap \underline{E_1}]$, on the other hand, is the prime event structure $(E_0 \times E_1, \leq', Con')$ where $(e, f) \leq' (e', f')$ iff $e' \leq_0 e$ and $f \leq_1 f'$; $X \in Con'$ iff $\pi_0 X \in Con_0$ implies $\pi_1 X \in Con_1$. Note that $\leq'$ is contravariant in its first argument. Therefore Axiom I cannot be preserved by the linear space construction.

There is an isomorphism between strict, linear functions from a prime algrbraic domain D_0 to a prime algrbraic domain D_1 and the ideal elements of the linear space from the prime event structure of D_0 to that of D_1. An ideal element R of the linear space $[\underline{E_0} \multimap \underline{E_1}]$ determines the function

$$\lambda x \in \mathcal{F}(\underline{E_0}).\{b \in E_1 \mid a \in x \;\&\; (a, b) \in R\},$$

which is linear (hence strict). The inclusion on the ideal elements gives the extensional order on the corresponding functions. On the other hand, any strict, linear function

$f : D_0 \to D_1$ determines the set

$$\{(a, b) \mid a \in D_0^p \;\&\; b \in D_1^p \;\&\; a \sqsubseteq f(b)\}$$

which can be regarded as an ideal element of the linear space from the prime event structure of D_0 to that of D_1.

Taking the ideal elements of the linear space as morphisms, we see that prime event structures form a monoidal closed category. In particular,

$$[\underline{E_0} \otimes \underline{E_1} \multimap \underline{E_2}] \cong \underline{E_0} \multimap [\underline{E_1} \multimap \underline{E_2}].$$

What is interesting about this category is that it contains a full subcategory of coherent spaces, which is also monoidal closed. Note that although the set of Scott continuous functions on coherent spaces (with the extensional order) is not a coherent space, the set of strict, linear functions is.

One wonders if the monoidal closed category of stable event structures can also be used to interpret classical linear logic [Gi87]. The sense is that it may not be so though we do not have a proof. To interpret classical linear logic a construction of convolution (or linear absurdity) is needed. Double convolution should give isomorphic structures ($A^{\perp\perp} \cong A$). However if one tries to follow Girard [Gi87] by taking $\multimap \perp$ as the convolution (which seemed not unreasonable), the double convolution of any stable event structure will be a coherent space (this is independently confirmed in [Co87])! Nevertheless, double convolution can help us understand the relation of the two different exponentials mentioned at the end of the previous section. We have

Proposition 5.1. For any coherent space X,

$$!_G X \cong (!_Z X)^{\perp\perp}.$$

This proposition says that if we first apply our exponential $!_Z$ to a coherent space X (the result may not be a coherent space) and then apply the double convolution (the result must be a coherent space), we get a coherent space isomorphic to the Girard's exponential $!_G$ of the original one. We leave the proof to the reader.

Finally, we remark that by providing a monoidal closed category of event structures and relating it to models of concurrency one hopes to open a new angle for a better understanding of the connection between linear logic and concurrency. It is beyond the scope of the present paper how a better understanding can thus be achieved. The exact connection between linear logic and concurrency remains to be seen.

REFERENCES

Be78 Berry, G. Stable models of typed λ-calculi, *Lecture Notes in Computer Science 62* pp. 72-88 (1978).

Co87 Coquand, T. Two models of intuitionistic linear logic in posets, manuscript, (1987).

Gi86 Girard, Jean-Yves. The system F of variable types, fifteen years later, *Theoretical Computer Science 45* pp. 159-192 (1986).

Gi87 Girard, Jean-Yves. Linear Logic, *Theoretical Computer Science 50* pp. 1-102 (1987).

GL86 Girard, Jean-Yves and Lafont, Y. Linear logic and lazy computation, manuscript, (1986).

Mc71 MacLane, S. *Categories for the working mathematician,* Springer-Verlag (1971).

Mi89 Milner, R. *Communication and concurrency,* Prentice Hall, (1989).

MM89 Narciso Marti-Oliet, Jose Meseguer. From Petri Nets to Linear Logic, manuscript, (1989).

NPW81 Nielsen, M., Plotkin, G., Winskel, G. Petri nets, event structures and domains, part 1, *Theoretical Computer Science* 13 (1981).

Sc82 Scott, D. Domains for denotational semantics, *Lecture Notes in Computer Science 140* pp. 571-613 (1982).

Se89 Seely, R.A.G. Linear logic, *-autonomous categories, and cofree coalgrbras, Categories in Computer Science and Logic, *Contemporary Mathematics*, 92 (1989) 371-382.

Wi87 Winskel, G. Event structures, *Lecture Notes in Computer Science 255* pp. 196-223 (1987).

Wi88 Winskel, G. An introduction to event structures, *Lecture Notes in Computer Science* 354 (1988) 364-397.

Wi88a Winskel, G. A monoidal closed category and its relation with categories for synchronizing processes, manuscript, (1988).

Zh89 Zhang, G.-Q. DI-domains as information systems, *Lecture Notes in Computer Science 372* (ICALP-1989, Italy), (1989) 773-788.

Zh91 Zhang, G.-Q. *Logic of Domains*, Birkhäuser Boston (1991).

Zh92 Zhang, G.-Q. DI-domains as prime information systems, *Information and Computation* 100 (2), 151-177 (1992).