On Continuous Time Agents [1]

S.Kasangian, Dipartimento di Matematica,Università di Milano
A.Labella, Dipartimento di Matematica, Università di Roma "La Sapienza".

Abstract

Continuous time agents are studied in an enriched categorical framework that allows for a comprehensive treatment of both the interleaving and the true concurrent paradigms in parallelism. The starting point is a paper by Cardelli, where actions have a duration in a (dense) time domain. More recent works are also briefly considered and some possible directions towards timed "true concurrent" processes are indicated.

Introduction

There is a wide spectrum of possible viewpoints in dealing with different aspects of parallel processes and computations , and the task of providing adequate models urged the deepening of the understanding of them. Historically, the outcome has been a variety of approaches based on different intuitions, ranging from Milner's CCS ([Mil83]), whose stress is on a notion of non-determinism over sequential processes, to "true concurrency" in the form, e.g., of Petri Nets (see [Rei85]). Many other interesting paradigms have been proposed like, for instance, Traces [Maz87], Event Structures [Win87], Concurrent Histories [DeM87], and Pomsets [Pra86]. Soon, the need for fertile comparisons among these paradigms and possibly for a unifying theory grew and gave rise to several attempts in this direction.

In a previous paper [KL90] we described a categorical model of synchronization between agents performing processes in *discrete time*. That is, we gave a *unified account* of CCS-like approaches. The basic ingredients were, for a start, a topos of "labelled trees" and a topos of "budding" trees constructed out of the first one. The latter turned out to be a monoidal category with respect to a tensor product which accounts for concatenation (temporal sequentialization) of processes, and has to be thought of as a "category of computations". Actual processes are then objects of categories enriched over it, which we called semantics.

---

[1]This work has been partially supported by the "Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo", CNR, Italy.

The present paper relies instead on a *more recent* categorical framework, introduced in [KLP90] which *"encompasses most of the models mentioned above and aims to provide a mathematical foundation to a* ***comprehensive*** *approach to concurrency, and which shares with systems à la CCS some metatheoretical assumptions, like the relative inaccessibility of computing agents and hence the crucial relevance of the notion of observability"*. In a slogan, the attempt of [KLP90] was somehow to "transform **metatheory** into actual mathematical **theory**".

Let us briefly recall that in [KLP90] we proposed a construction based parametrically on many *systems of observers* and their related *observations*. By adjusting the observers we got the different models of parallel computations we wished to compare. Observers are used to analyze and/or construct trees of computations, which in their turn are viewed as *categories enriched in a bicategory (actually a 2-category) of observers*. We emphasised the notion of "experiment performed by an observer", and the idea of associating *two kinds of information* to the experiments: an "intrinsic" information (*extent*) and a relational information (*agreement*). Finally, trees of computations (viewed as enriched categories) are gathered in a monoidal category, like the one described in our earlier approach ([KL90]), in which categories of processes are actually enriched. Namely, trees of computations represent the transformations that a process may undergo while getting to one of its future states. Hence, the novelty was the introduction of a new "bottom" level, the observers, in this chain of enrichments, which allows for a sharper analysis in terms of observability.
Since our models are parametrized by the *systems of observers*, we were able to compare them and to exhibit intrinsic relationships among the different kind of semantics involved.
The first example we addressed in [KLP90] was the theory of SCCS as introduced in [Mil83] (and this covers essentially also TCSP as presented, for instance, in [BHR84]). In SCCS the rules describing the agents behaviour are labelled by atomic actions. This corresponds to the explicit assumption that one is able to observe *only one action at a time*. The implicit assumption is that time is supposed to be discrete and linearly ordered. Accordingly, we adopted a system of *linearly ordered* observers . At the other extreme of our range, we had examples related to *true concurrency* as, for instance, Mazurkiewicz traces [Maz87] and Event Structures [Win87], which forced the assumption of a system of *partially ordered* observers.

In the present paper we want to make explicit the role played by the (implicit) notions of time inherent to our models, considering for instance the natural extension of the system of observers corresponding to a discrete and linearly ordered time to *continuous time observers*, obtained assigning a *duration* to actions. This approach essentially covers also a variety of treatments of *timed process algebras*, based on instantaneous actions with a delay or similar features. Although we provide a sketchy starting point for *timed concurrent processes*, time will be always assumed to be totally ordered, since the ability of observing more than one action at a time is due rather to the complex and ramified nature of the observers involved.
In fact we focus on the special case of *continuous time* and therefore we are led to consider trees as built starting from *paths* (rather than from nodes, like in [KaL90]). This different view is useful also in the particular case of discrete time .

Paths are to be thought of as *copies of the observers*. In the case of unlabelled trees, that can be viewed as categories *enriched in a suitable bicategory* constructed out of the topological space induced on $\mathbb{R}^+ \cup \{\infty\}$ by the "segment" topology. Concatenation is then a new notion of *tensor product* on the bicategory. Observers are in this case simply *durations*. Considering instead *labelled* continuous trees as behaviours of agents (and also as computations between them), we are led to a more general theory that *disregards the topology on the extended reals* and starts from a mere meet-semilattice structure to construct a suitable base bicategory of observers for real time agents, in order to model parallelism, following rather closely the point of view of Cardelli [Car82] .
After a heuristic introduction to our theory of experiments (section**1**), we provide a rather self contained section on enriched categories and sheaves, in order to be able to deal with Cauchy-completeness. Section **3** is an illustration of the theory for unlabelled trees. In section **4** we treat continuous labelled trees, in detail, modelling Cardelli's real time agents and commenting on more recent work. The last section, of rather tentative nature, is dedicated to an approach to timed Mazurkiewicz traces.

Let us conclude this introduction recalling some of the basic notions of Category Theory that turned out to be relevant for parallel processes and are related to our approach.
In the first place, let us mention *Monoidal Categories* (see e.g. [EiK65] , [Kel82]), which have been used by Montanari and Meseguer to provide an algebraic description of Petri nets (see e.g. [MeM88] and [DMM89]) and even earlier by Benson ([Ben89]) in his approach to non-determinism and concurrency which applies Hopf Algebra theory in a semiring-theoretical context.
Bicategories, introduced by Bénabou (see [Bén67]), have been used for rewriting systems by Eilenberg, Street and others . Notice that, in fact, the bicategories considered are actually *2-categories*.
Enriched Categories, introduced by Eilenberg and Kelly in [EiK65] and by J.Bénabou [Bén65] (see also [Kel82]) were used by Betti and Kasangian in Automata Theory, to describe sequential and tree automata (see [Bet80] and [KKR83]) and later by the present authors in [KLP87] and [KaL90] to represent processes as categories enriched in monoidal categories of trees, representing *computations*. Another application of enriched category theory is provided in [CCM90]
The crucial notion of *category enriched in a bicategory* was introduced by Walters [Wal81] in order to show that sheaves on a topological spaces are equivalent (as a category) to skeletal, symmetric *Cauchy-complete* categories enriched on a bicategory constructed from the space itself. Further, Betti and Kasangian showed that tree automata are categories enriched in a bicategory [BeK85] and very recently this notion turned out to be strongly related to Rosenthal's *quantaloids*.

We would like to thank the referees, David Benson, GianCarlo Mauri, Miranda Mowbray, David Murphy, Alberto Pettorossi, Nicoletta Sabadini and Sebastiano Vigna for valuable comments and helpful discussions.

§1 A heuristic approach to the theory of experiments

As we observed in the introduction, trees can be viewed as the unfolding of agents. Following an old tradition in language theory, we view such trees as the "record" of several "experiments" (observations) performed by observers *on* the agent, which of course is supposed to be reinitializable.

Observers are to be thought of as particular agents, which are *fully known*, while agents in general are considered, as usual, to be black boxes. Further, they are *virtual*, in the sense that they represent all the individual possible behaviours of an *actual* observer in the usual sense (see [Hen88]).

An experiment on an agent consists in its interaction with a particular observer. This interaction can be viewed as a synchronization à la Hoare between them.

The present *theory of experiments* is parametric with respect to a class of calculi for concurrency, like process description languages, real time agents, Mazurkiewicz traces, event structures, pomsets etc., according to the nature of the observers.

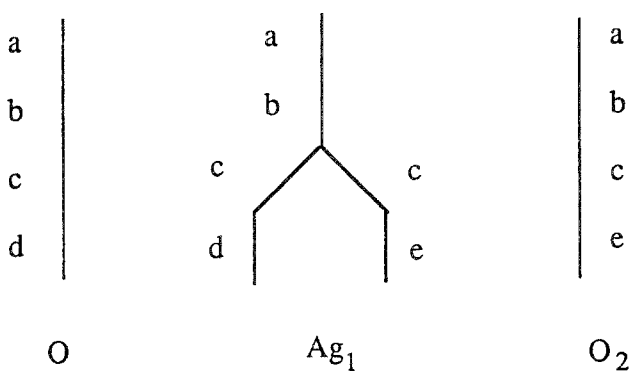
A particular calculus is then an admissible parameter of our theory if it satisfies at least the following requirements, namely, if it is possible to define:

i) a subset of particular computing agents which can be viewed as a *system of observers,*

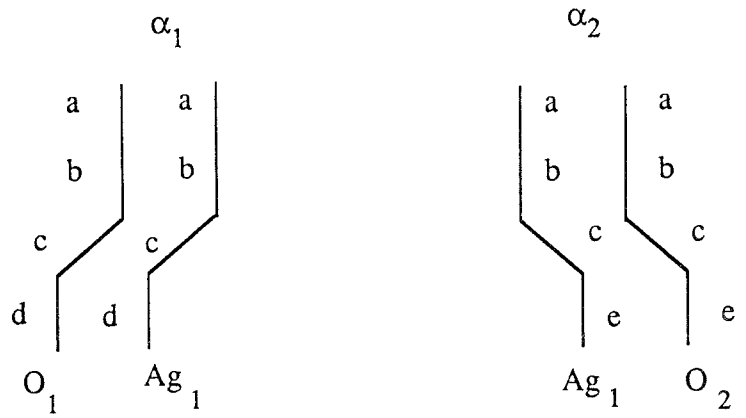
ii) a Hoare-like synchronization, denoted by ||, together with an agents-reduction relation (whose reflexive and transitive closure is denoted by $\rightarrow^*$), by which an agent-observer pair is reduced to a new agent-observer pair,

iii) an *irreducible* observer, denoted by NIL, such that for any agent ag_1 there does not exist any agent-observer pair p, different from $ag_1 \,||\, NIL$, with $(ag_1 \,||\, NIL) \rightarrow^* p$.

Example. In a process description language setting, let us consider an agent and two observers:



The possible experiments are then:



A successful experiment is characterized by the fact that the observing agent gets reduced to the irreducible observer NIL. Thus, a successful experiment, say α_1, on the agent ag_1 performed by the observer o_1 can be viewed as the reduction: $(ag_1 \parallel o_1) \rightarrow^* (newag_1 \parallel NIL)$. We say that α_1 is the successful experiment from the agent ag_1 to the agent $newag_1$. Agents ag_1 and $newag_1$ will also be called the *initial* and *final* agent of the experiment α_1.

Informally, a successful experiment "consumes" the observer. Records of successful experiments will always be either a root-to-leaf path in the tree representing the agent, or one of its prefixes. In our example, for instance, an observer "ab" is able to perform a successful experiment on ag_1, while "abcdefgh" is not. A *system of observers* is a partial order under the prefix relation. For instance, $ab \leq abcd$. This expresses the obvious fact that observer ab is able to "explore" ag_1 to a lesser extent than abcd= o_1. Further, two observers like o_1 and o_2 have a set of common prefixes, namely {a, ab, abc}, which represent the possible "agreements" between experiments α_1 and α_2 . Hence, all the information about a system of observers are concisely expressed by a 2-category, whose objects are the observers, 1-cells between two observers are their common prefixes, and 2-cells are given by the prefix ordering.

Given such a system (2-category) of observers, it is then possible to "explore" and reconstruct the behaviour of an agent through experiments, provided one takes into account both the intrinsic information given by an experiment (extent) and the relational one (agreement). Namely, the intrinsic information (extent) of the experiment α_1, denoted by $e(\alpha_1)$, is just o_1: that is, the observing agent itself, an object of the base 2-category. We can say that the computation with extent o_1 can be performed by the agent ag_1, which hence becomes the "new" agent $newag_1$. An unsuccessful experiment does not determine any computation.

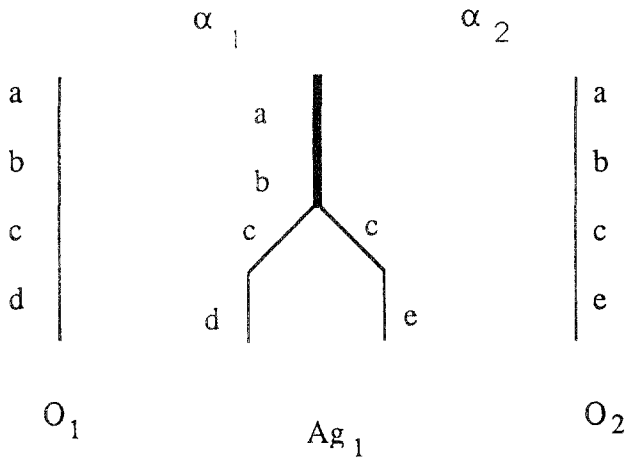
Since in general more than one successful experiment may be performed from one agent to the second, we associate with any two such experiments, say α_1 and α_2, the relational information (agreement) denoted by $\alpha(\alpha_1,\alpha_2)$ which is a 1-cell from o_1 to o_2 in the base 2-category.

The agreement expresses for how long two experiments have been inducing the same sequence of "observations" on the observed agent ag_1. Notice that the notion of agreement provides some informations about the internal structure of the agents and their evolution

during the experiments. Where the agreement ends we are obviously in the presence of a non-deterministic choice or a *conflict* (see[KLP90] for the connections with event structures).

The set of all possible computations from ag_1 to $newag_1$ turns out to be structured by the agreement between the corresponding successful experiments. Such a structured set is called *structure of computations* and is modelled as a category based on the 2-category of observers. A structure of computations has a tree-like representation, where a root-to-leaf path is a computation and *is labelled by its extent.* The agreement between any two computations determines the *fusion of the corresponding paths*, as it happens in a tree.

Let us have a look at the following example of agreement:



The agreement between α_1 and α_2 is "abc", i.e., still an observer.
Notice that two experiments with the same extent, in total agreement, must be identified. This requirement corresponds to the *skeletality* of the categories involved.
We may assume that there are agents that are *not* "observably decomposable" and behave in this sense as atomic actions. Observers are *indecomposable agents*, and are glued on common prefixes, where the various observations actually *agree.*
Summarizing, we have a three level categorical semantics, that is parametric w.r.t. the system of observers. The 'level 0' is represented by the 2-category of observers. The structures of computations are categories enriched in the 2-category of observers and are the objects of the monoidal category which is the 'level 1'. Processes are categories enriched in the monoidal category just mentioned. They are the 'level 2' of our three-level framework .
This apparently complicated "tower of enrichments" has in fact interesting payoffs. For instance, we can express in our framework also a *multi-level notion of observational equivalence among agents*. We have a lower level, given by the system of observers: we consider equivalent those agents whose computations cannot be distinguished by the experiments. Further, we have higher levels, which are provided by suitable equivalences introduced among structures of computations, and therefore, among agents.

For example, in the case of process description languages, like SCCS, we may assume that agents which differ for the duration of their actions are equivalent at lower level, because the usual SCCS-like observers are not able to distinguish them, while equivalences like $ag_1+ag_1 = ag_1$ can be introduced at higher levels (see [DeL91] and [KaP]).

§2 Preliminaries on Enriched Categories and Presentations of Sheaves

In [KaL90] we described computations among agents as trees and we defined a topos of trees, viewed as sets with an ultimately constant endofunction, and hence toposes of *labelled trees* and of *"budding trees"*. Our topos of (unlabelled) trees is equivalent to Bénabou's topos $\mathbf{S}^{\mathbf{N}^{op}}$ of presheaves (actually sheaves) over a topological space constructed out of the order of $\mathbf{N}$ via the segment topology [Bén84] . So it is very natural to define analogously a topos of continuous trees $\mathbf{S}^{\mathbb{R}^{+op}}$, where $\mathbb{R}^+$ are the extended reals with the segment topology.
Let us also recall that in [Wal81] R.F.C. Walters observed that sheaves over a topological space (or a locale) can be viewed as *Cauchy-complete* categories enriched in a suitable bicategory constructed out of the topological space. We give a brief outline of [Wal81] in order to apply the theory to computations "between" agents, viewed as continuous labelled trees. We will describe first unlabelled trees, viewed as *presentations* of sheaves on $\mathbb{R}^+$. Let us establish a few preliminary notions and terminology.

A bicategory **B** is a set of objects Ob(**B**) together with, for each pair of objects (a,b), a category **B**(a,b). For any (a,b) the objects of **B**(a,b) are called morphisms of **B** (or 1-cells of **B**), and they will be collectively denoted by morph(**B**). Analogously, for any (a,b) the arrows of **B**(a,b) are called 2-cells of **B**.
For each object a of **B** an identity 1-cell in **B**(a,a) is given, and a composition functor is given for each triple of objects (a,b,c). Further, for each 4-tuple of objects there is an associativity isomorphism, and for each pair of objects there is a left and right identity isomorphism satisfying suitable coherence conditions ([Ben67, p.5-6]. In particular, if **B**(a,b) is a poset (qua category), the bicategory is called *locally posetal* and is actually an **Ord**-category. In this case the coherence conditions are trivially satisfied because all the diagrams involved commute.

Remark. If composition of morphisms is strictly associative and the identities are strict w.r.t. composition, the bicategories are called 2-categories [KeS74]. Categories in their turn are special 2-categories with only identity 2-cells. Thus, with respect to the definition of a category, a bicategory generalizes the *set* of morphisms between any two objects to a *category* of morphisms (that is, a category of 1-cells).

Let us now present a simple but important example of bicategory. This example motivated also the original notion of categories based on a bicategory [Wal81], which plays a central role in our categorical framework (see Definition 2).

Example 1. Let us consider the lattice **T** of open sets of a topological space T. From **T** we can construct a bicategory (to be called **B(T)** or simply T) whose objects are the elements of T. For any two objects u and v in **T**, the category **T**(u,v) has as objects (that is, 1-cells) the open sets contained in $u \cap v$ and as morphisms (that is, 2-cells) the inclusions among open sets. For any two 1-cells we define the composition as the intersection of open sets, and for any two 2-cells the composition is given by the transitivity of the inclusion.

Let us now give a formal definition of the kind of posetal bicategories (actually, 2-categories) we will consider in this paper.

Definition 1. Given a meet-semilattice $(B,\leq,\wedge,\perp)$, where B is the underlying set, $\leq$ is the partial order over B, $\wedge$ is the meet operation, and $\perp$ is the bottom element, a semilattice-bicategory (called sl-bicategory, for short) **B** consists of:
i) a set of objects Ob(**B**), which coincides with B,
ii) for each pair of objects a and b, a poset **B**(a,b) (viewed as a category), defined by $\{k \mid k \in B \text{ and } k \leq a \wedge b\}$. For each pair of objects a and b, the objects of **B**(a,b) are called 1-cells, and the arrows of **B**(a,b) are called 2-cells,
iii) for each triple of objects a, b, and c, an associative operation from pairs of 1-cells in $\mathbf{B}(a,b)\times\mathbf{B}(b,c)$ to a 1-cell in **B**(a,c), called composition of 1-cells, and defined by the meet operation $\wedge$,
iv) for each object a, an identity 1-cell, denoted by $\mathbf{1}_a$, in the category **B**(a,a), which is a itself (that is, the greatest object in **B**(a,a)). $\mathbf{1}_a$ is a left and right identity for the composition of 1-cells,
v) for each pair of 1-cells f and g, *at most* one 2-cell, defined by the relation $f \leq g$,
vi) for each pair of objects a and b and for each triple of 1-cells in **B**(a,b) an associative operation, called vertical composition of 2-cells, provided by the transitivity of the partial order $\leq$.

For each object a, there exists an identity 2-cell, denoted by $\mathbf{i}_a$, from $\mathbf{1}_a$ to $\mathbf{1}_a$, which is provided by the reflexivity of $\leq$. $\mathbf{i}_a$ is a left and right identity for the vertical composition.
By definition, for each pair of objects a and b we have that: **B**(a,b)=**B**(b,a).

Definition 2. Given an sl- bicategory **B**, a *B-category* **X** is a triple $\langle X,e,\alpha\rangle$ given by set X together with an *extent* function $e:X \to \text{obj } \mathbf{B}$ and an *agreement* (or *glueing*, or *distance*) function $\alpha(-,-): X\times X \to \text{morph } \mathbf{B}$ (often denoted like an homset $\mathbf{X}(-,-)$)satisfying:
i) $\alpha(x_1,x_2): e(x_1) \to e(x_2)$
ii) $1_{e(x)} \leq \alpha(x,x)$
iii) $\alpha(x_1,x_2)\cdot\alpha(x_2,x_3) \leq \alpha(x_1,x_3)$
A **B**-category is *symmetric* if $\mathbf{X}(x_1,x_2) = \mathbf{X}(x_2,x_1)$. Notice that this notion makes sense since **B** is an sl-bicategory.
A symmetric **B**-category is called *skeletal* iff for any two objects x_1 and x_2 in **X** we have that: if $e(x_1)= e(x_2)= \alpha(x_1,x_2)$ then $x_1= x_2$

Going back to **Example1**, in Walters' description symmetric and skeletal **T**-categories correspond to presentations of sheaves, and "are" actually sheaves iff they are Cauchy-complete (see below).
B-categories can be gathered in a bicategory in a few different ways. The most relevant for the present paper is maybe **B**-Cat, whose objects are **B**-categories and morphisms are **B**-functors.

Definition 3. A *B-functor* $f:X\to Y$ is a function on the set of objects of X and Y satisfying:
i) $e(f(x)) = e(x)$
ii) $X(x_1,x_2)\leq Y(fx_1,fx_2)$

Some very important results have been obtained by considering different kind of morphisms between **B**-categories.

Definition 4. A *bimodule* ([Law73], [Wal81]) $\Phi:X\multimap Y$ is a function $\Phi: X\times Y \to$ morph **B** satisfying (for any x,x' in X ; y,y' in Y)
i) $\Phi(x,y): e(x) \to e(y)$
ii) $X(x',x)\cdot \Phi(x,y)\leq \Phi(x',y)$
iii) $\Phi(x,y)\cdot Y(y,y')\leq \Phi(x,y')$
The bicategory whose objects are **B**-categories and morphisms bimodules (also called *modules*) is denoted by **B-mod** (see [Law73] and [Wal81])

Definition 5. Let $\Phi:X\multimap Y, \Psi:Y\multimap X$ be bimodules, they are *adjoint* iff the following conditions hold:
i) $X(x,x')\leq \sum_y \Phi(x,y)\cdot \Psi(y,x')$
ii) $Y(y,y')\geq \sum_x \Psi(y,x)\cdot \Phi(x,y')$
Of course a **B**-functor $f:X\to Y$ yields a pair of adjoint bimodules $f^*:X\multimap Y$, $f_*:Y\multimap X$ defined by $f^*(x,y)=Y(fx,y)$ and $f_*(y,x)=Y(y,fx)$.

Definition 6. A **B**-category is *Cauchy-complete* if any pair of adjoint bimodules arises from a **B**-functor.

Walters' result is then ([Wal81][Wal82]) that if H is a locale, the category of *symmetric and skeletal* Cauchy-complete **Rel**(H)-categories is equivalent to the category of sheaves on the locale H.
(*) $\mathbf{Sh(H)} \approx \mathbf{SkSymRel(H)\text{-}Cat_{cc}}$
An equivalent statement is the following one (where we call Map**B** the bicategory of **B**-categories and *adjoint pairs of modules*, or "maps", that is arrows with a right adjoint) :
(**) $\mathbf{Sh(H)} \approx \mathbf{Map(SymRel(H)\text{-}mod)}$
The difference is that in (*) the objects are Cauchy-complete categories, which correspond to sheaves, while in (**) the objects are *all* the categories, to be thought of as *presentations* of sheaves: there are sections (but not all of them) and some glueing data among them are assigned, but, for instance, not all restrictions have to be defined (so a **B**-category is not even

a presheaf). Further, in (*) morphisms are **B**-functors and in (**) they are adjoint pairs of modules.

§3 Continuous trees as presentations of sheaves

Let us now apply the theory of the previous section to the example of *unlabelled* continuous trees.

The extended reals $\mathbb{R}^+ = \mathbb{R}^+ \cup \{\infty\}$ are a topological space with the topology having as open sets the intervals of the form $[0,a)=\{x \text{ in } \mathbb{R}^+ \mid x< a\}$, called the "segment" topology.

Continuous trees (actually, forests) are just sheaves on $\mathbb{R}^+$ or equivalently, as it is easy to show, presheaves of the form $S^{\mathbb{R}^{+\,op}}$. So they admit a description as small categories enriched in $\mathbf{B}(\mathbb{R}^+)$.

Let us draw a picture of $\mathbf{B}(\mathbb{R}^+)$ and a picture of **X**, a category enriched over it

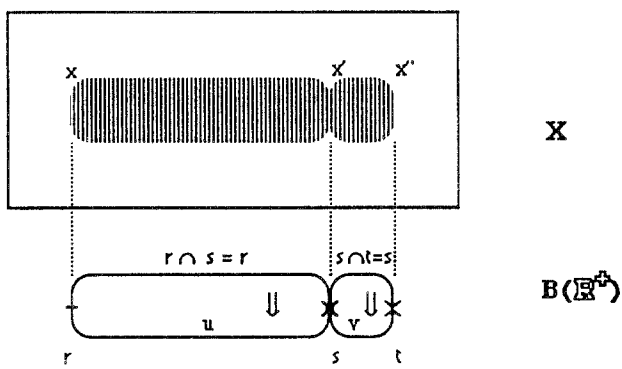


Fig.1 A category enriched in $\mathbf{B}(\mathbb{R}^+)$

where r, s, t are segments of $\mathbb{R}^+$. The objects x, x', x'' of **X** are copies, respectively, of r, s and t, while u, v are their agreements (glueing data), which can be viewed as *sections*,

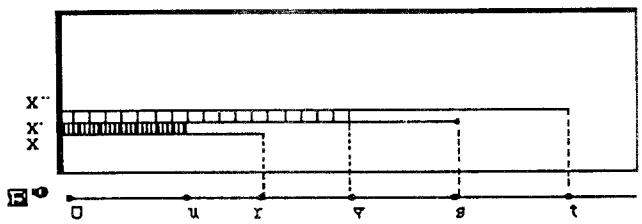


Fig.2 The sections of the presentation of a sheaf and their glueing data

or, better

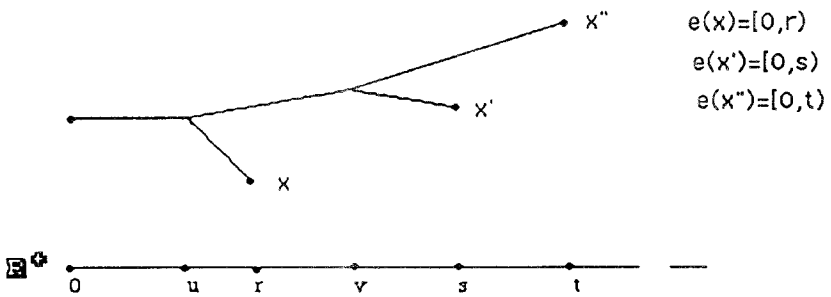


Fig.3 Sections are *paths* and glueing data are *agreement data* : it's the presentation of a *tree*

So, sections with extent, say r, are just *paths of length* r. A $\mathbf{B}(\mathbb{R}^+)$-category **X** should be thought of as a presentation of a sheaf, but represents indeed a continuous tree. The well known contradiction between *presentations* and *invariant presentations,* like the one between sketches and theories, shows up once more. In fact, $\mathbf{B}(\mathbb{R}^+)$**-Cat** is not equivalent at all to sheaves on $\mathbb{R}^+$ which, let us recall, are rather equivalent to **Map(SymRel** ($\mathbb{R}^+$)**-mod)**. So, to get all possible restrictions of any section (i.e. all possible prefix-subpath of any path) one has to perform the Cauchy-completion construction. However, and this will be far more evident in the next section on *labelled* trees, in order to model continuous systems (or agents), we may assume the existence of actions with duration which are *not* "observably decomposable" and behave in this sense as atomic actions. In the framework of [KLP90], a continuous tree is built by patching together paths (sections) that are copies of elementary "observers" which realize observations that can be seen as synchronizations à la Hoare. Observers are *indecomposable intervals*, and are glued on common prefixes, where the various observations actually *agree*, i.e. are compatible, or *observably indistinguishable.* That's why the glueing of two paths is called *agreement* .

The notion of tree morphism $X \to Y$ we need must require:
i) a function on the *paths* (not any longer on the nodes, like in [KaL90]) of **X**

ii) which preserves the length of paths and
iii) is not decreasing w.r.t.the amount of glueing (agreement) of the paths.
But these conditions are expressed just by
i) f: ob**X**→ob**Y**
ii) e(f(x)) = e(x)
iii) **X**(x_1, x_2) ≤ **Y** (fx_1, fx_2)
i.e. *precisely* the notion of **B**-functor.

Remark. Notice that we did not really need the topological structure of the *extended* non negative reals just to present continuous trees. The relevant thing is the presence of a meet semilattice and hence of a sl-bicategory, on which trees turn out to be enriched. Cauchy completions of course require the existence of local colimits stable by composition of 1-cells, and this might not be the case. The base bicategory must then be suitably completed. However, if one is interested in presentations and assumes an *atomicity* of actions, this is not a problem. The "granularity" of actions depends of course on the choice of the base bicategory and the abstraction *and refinement* problems can be dealt with suitable homomorphisms of bicategories called *changes of base* (see [CKW87]).

§4 Behaviours of real time agents as continuous labelled trees

In this section we study continuous time agents, mainly following Cardelli [Car82], who explicitly models a "dense" timed version of Milner's SCCS [Mil83]. In the "discrete time" case agents must perform actions of unitary duration, while in the continuous time case we assume that an elementary action can be performed with different durations Hence, in the following definition, we introduce a notion of *observer* which corresponds to Cardelli's *deterministic agent* and extends the analogous notion for the discrete time case, namely a word in a free monoid (see e.g. in [KLP90]). Note that, differing from Cardelli, we admit actions with zero duration.
Let $\mathbb{R}^+$ denote the set of non negative real numbers (notice: not *extended*).

Definition 7. Given an alphabet A of atomic actions, a *partial piecewise constant function* (*pc-function*, for short) f is a function from $\mathbb{R}^+$ to A defined on a *bounded* interval $[0,t) \subseteq \mathbb{R}^+$, such that, for any a∈ A, $f^{-1}(a)$ is the union of *finitely* many intervals of the form [r,s), with r and s non-negative reals.

This definition ensures that in every bounded interval of $\mathbb{R}^+$ only "finitely many things can happen", preventing undesirable situations like "Zeno machines" (see for instance [JoG85]).

Proposition 1 The set CA of partial piecewise constant functions is a meet-semilattice:
i) f≤g iff dom(f) ⊆dom(g) and if x ∈ dom(f) then f(x)=g(x),

ii) $f \wedge g$ is defined as follows:

- $dom(f \wedge g) = [0,t)$ where t is such that:

$\forall x<t$ both f(x) and g(x) are defined and f(x)=g(x), and

if f(t) and g(t) are both defined then $f(t) \neq g(t)$,

- if $x \in dom(f \wedge g)$ then $(f \wedge g)(x) = f(x)$

iii) the bottom element $\perp$ is the empty function defined on the empty interval [0,0). ¶

The meet of the pc-functions f and g of Figure 4 is the pc-function h such that $dom(h)=[0,t_2)$, $h^{-1}(a) = [0,t_1)$, $h^{-1}(b) = [t_1,t_2)$.

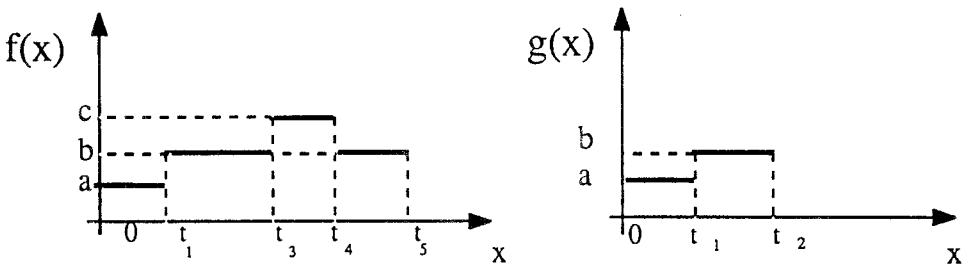


Figure 4 Two piecewise constant functions f and g. (Assume that if $i<j$ then $t_i<t_j$).
$dom(f)=[0,t_5)$, $f^{-1}(a)=[0,t_1)$, $f^{-1}(b)=[t_1,t_3) \cup [t_4,t_5)$, $f^{-1}(c)=[t_3,t_4)$.
$dom(g)=[0,t_2)$, $g^{-1}(a)=[0,t_1)$, $g^{-1}(b)=[t_1,t_2)$.

As a consequence of the proposition above, we can readily disregard the topological structure on the extended reals: what really counts is the introduction of a *base* sl-bicategory.

Proposition 2 **CA** $=(CA,\leq,\wedge,\perp)$ is an sl-bicategory. ¶

CA is the *system of observers for real time agents*. A system of observers (an sl-bicategory) **A** for discrete time agents can be recovered as a special case by considering each sequence of actions, that is, a word in A*, as a partial pc-function from the *natural numbers* to A, defined on an initial segment [0,n) for some n. In this sense **A** turns out to be a discrete version of **CA** (see [KLP90] for the details). Notice again that *observers for real time agents* are special agents themselves, namely Cardelli's *deterministic agents*. In his operational semantics for deterministic agents Cardelli has a left prefix *deterministic* operator a[t]:- , that we obtain as a special case of concatenation of two pc-functions.

Given two pc-functions $f: [0,s) \rightarrow A$, and $g: [0,t) \rightarrow A$ they can be concatenated as follows:

$$(f.g)(x) = \begin{cases} f(x) & \text{if } x<s \\ g(x-s) & \text{otherwise} \end{cases}$$

In fact, notice that the concatenation we defined is *not symmetric* and extends to a *global tensor product* on the 2-category **CA** which is more general than the one defined by Carboni and Walters [CaW87]. By a global tensor product we mean an *associative morphism of bicategories* $\otimes: \mathbf{B} \times \mathbf{B} \rightarrow \mathbf{B}$ with an identity I. Notice that in the present example the tensor product is *strictly* associative, providing a special case of the *semistrict monoidal 2-categories*

studied by Kapranov and Voevodsky [KaV91] in connection with Quantum Groups. Finally, we recall that our tensor product is not *symmetric*, in that accounts for concatenation of processes.

Proposition 3. The concatenation of pc-functions defines a global tensor product $\otimes$: **CA**×**CA** $\rightarrow$ **CA** on the sl-2-category **CA** .

Proof. Straightforward. The identity of the tensor product $\otimes$ is the empty function $\perp$.

¶

The remaining reduction rules of Cardelli's operational semantics for *deterministic* agents include the rule for the idle process (which is implicit in our setting) and the rule ($\times \rightarrow$) for synchronization of agents, that we will treat as a special case in our account on non-deterministic agents. The meaning of the rule (a[][]$\rightarrow$), according to Cardelli, is that: "after an interval t, a[t+u]:p has only reached a[u]:p"[Car82]. Cardelli observes further that "it is possible to split actions at arbitrary points, but this is done consistently so that the final outcome remains the same". Therefore, this rule and its important consequence, the *Density Lemma*, ensure the existence of all possible decompositions of a timed action, hence of any observer (deterministic agent). We will see later, again as a special case of non-deterministic agents, the connections between Density Lemma and the Cauchy-completeness of enriched categories and hence the correct applications of sheaves. Finally, let us notice that in order to extend the use of the rule ($\times \rightarrow$) to agents with the same overall duration but with different decompositions the use of the rule (a[][]$\rightarrow$) is necessary .

Other important features of Cardelli's deterministic agents are the *Refinement Lemma* and the *Smooth Equivalence*. Informally, actions which are "similar", i.e. differ only in their decompositions in time, can be considered equivalent. This gives rise to a congruence and to normal forms that correspond to the "coarsest" granularity of the actions. For instance, a[2]:a[5] becomes a[7]. It is interesting to observe that we can quotient out the base 2-category, getting observers *only* in normal form.

Let **SkSymCA-Cat** be the category of *skeletal symmetric* categories based on **CA**. **SkSymCA-Cat** can be viewed as the category of *all trees whose root-to-leaf paths are labelled by partial piecewise constant functions*. Also the agreement between any two paths is a partial piecewise-constant function (see Figure 5 below). Let us avoid heavy notations and let us rename **SkSymCA-Cat** as **CTrees** (short for Continuous Trees).

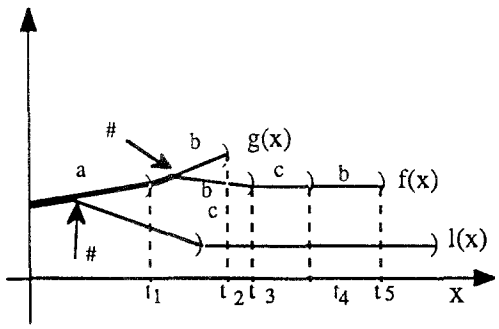


Figure 5 An object of **CTrees**, that is, a continuous tree with 3 paths: f, g and l. The agreement between f and g is smaller than h=f∧g, because of the choice point in #.The same thing happens with the agreement between l and g

The category **CTrees** provides an adequate model for the computations performed by real time agents. According to Cardelli's terminology, its objects are to be thought of as the unfolding of *non-deterministic agents* : actually, they are at least *vertically* non-deterministic in the sense that the branching structure models non-deterministic choice. Horizontal non-determinism models indeterminacy in the duration of actions. Let us quote Cardelli: *"We can imagine the behaviour of an agent as a trajectory in the plane, with time on the x axis and the action monoid on the y axis. This explains the sense of the adjectives 'horizontal' and 'vertical'."* ([Car82]). Figure 5 is indeed an example of this metaphor .
Horizontal non-determinism, however, is missing. The point is again that objects in **CTrees** are trees that are to be thought of as "presentations of sheaves", so that they do not have all restrictions, which would allow for both stretching of action durations and a finer atomicity w.r.t. time. However, as we observed at the end of the previous section, we may assume the existence of actions with duration which are *not* "observably decomposable" and behave as atomic actions. We can deal with the "granularity" of the observations by choosing atomic actions possibly with short duration. This is of course possible since *all* the observers are in **CA**. Hence, objects in **CTrees** can be considered as a relatively new model with respect to Cardelli's. As a particular case, the coarsest granularity corresponds to the choice of observers in normal form w.r.t. smooth equivalence.
Further, in the context of presentations we can restrict ourselves to *finite* categories, i.e. with a finite number of sections which, let us recall, correspond to experiments.

The concatenation $\underline{\otimes}$ of continuous trees in **CTrees** can be defined by lifting to **CTrees** the tensor product defined on **CA**.

Proposition 4 **CTrees** is a monoidal category, with tensor product
$\underline{\otimes}$:**CTrees** × **CTrees** → **CTrees**.
Proof. The tensor product is defined on the objects as follows:
let **X** and **Y** be objects of **CTrees**. Then :
$\mathbf{X} \underline{\otimes} \mathbf{Y} = \mathbf{X} \times \mathbf{Y}$
$e(x,y)=e_1(x) \circ e_2(y)$

$$\alpha((x,y),(x',y')) = \begin{cases} \mathbf{X}(x,x') \circ \mathbf{Y}(y,y') & \text{if } \mathbf{X}(x,x') = e(x) = e(x') \\ \mathbf{X}(x,x') & \text{otherwise} \end{cases}$$

The identity for $\otimes$ is the tree **CNIL**=<{p},e,a> where $e(p)=\alpha(p,p)=\perp$.
As for morphisms, if $f: X \to X'$ and $g: Y \to Y'$
$f \otimes g: X \otimes Y \to X \otimes Y'$ is given by $(f \otimes g)(x,y)=(f(x),g(y))$ and
$(f \otimes g)\ \alpha((x,y),(x',y'))= \alpha((f(x),g(y)),(f(x'),g(y')))$. Then from the functoriality of f and g and from the skeletality of the categories involved it follows that:
$\alpha((x,y),(x',y')) \leq \alpha'((f(x),g(y)),(f(x'),g(y')))$, i.e. $f \otimes g$ is functorial. The rest of the proof is straightforward. ¶

This result may look slightly surprising from the point of view of Enriched Category Theory. Indeed, in the case of categories enriched on a monoidal category, the symmetry of the base is necessary to get the "middle four interchange" law which ensures the possibility of defining a tensor product at the **V-Cat** level. Here, the corresponding construction comes just from the (lax) functoriality of the tensor, and this is due to the peculiarities of the above definition.
Hence, **CTrees** is a *monoidal category of computations* and continuous time agents are to be thought of as categories enriched in **CTrees**. Products, sums etc. exist in **CTrees** and have nice interpretations. In particular, the sum models the + operator as a non-deterministic choice.
Let us recall that in this "tower of enrichments", from level 0 (the 2-category **CA** of observers) to level 1 (the monoidal category **CTrees** of unfolding of agents, or computations) and level 2 (**CTrees-Cat**, the category of continuous agents), each level subsumes the previous ones. So, an observer is an object of **CA**, but it can be seen as a one-object **CA**-category, i.e., as a tree with a single branch or, better, as the unfolding of a deterministic agent in the sense of Cardelli. Further, all the trees of **CTrees** can be viewed as particular **CTrees**-categories, as we showed in [KaL91] and [KLP90] for the discrete time case. This allows for a comprehensive treatment of the various kinds of observational equivalences, according to the context. This has been done in [KaP91] for strong equivalence in the SCCS case, and more in general in [DeL91] and can be extended to the continuous time case: this is however beyond the scope of the present account.
In [Car82] two prefixing operators for continuous trees are defined: we met already the deterministic prefixing a[t]:- (prefixing a continuous tree by the action a performed for duration t). It can be modelled both in **CA** and in **CTrees** as a restriction of the tensor products $\otimes$ and $\otimes$ defined above. In particular, in **CTrees** the left operand is the tree <{p},e,α>, where both e(p) and $\alpha(p,p)$ are the pc-function assuming the value a in [0,t). This function will be denoted by $f_a(t)$ in the sequel.
Cardelli introduces further the *horizontal* non-deterministic prefixing a(t):- (prefixing a continuous tree by the action a performed for an *arbitrary duration* x, $0<x\leq t$).
It can be defined once more as a restriction of the tensor product $\otimes$ on **CTrees**, but now the left operand is not any longer the one-object category <{p},e,α>, but rather a category with an *infinite* set of objects p_x , where e_x is a partial *constant function* (that is, a pc-function which takes only one value) defined on [0,x), $x \leq t$, with value a. In other words, we need

restrictions . Trees in **CTrees** with this property model properly Cardelli's non-deterministic agents.
We need to recall some terminology from section **2** , namely the notion of *map*, i.e. an arrow with right adjoint. Let us describe maps in the 2-category **CA**. Suppose we are given a map h: f $\rightarrow$ g, i.e. a common prefix of the pc-functions f and g with a right adjoint l:g$\rightarrow$ f such that f $\leq$ h $\wedge$ l and l $\wedge$ h $\leq$ g (unit and counit of the adjunction). But the unit forces f $\leq$ l and f $\leq$ h, f = h = l, while the counit guarantees that f $\leq$ g. So, to consider a map h: f $\rightarrow$ g means that f $\leq$ g and that we are taking the maximal common prefix, i.e., f.
Restrictions in a **B**-category have been introduced by Betti and Walters in [BeW85] and in [BeW87] , where they are called "substitutions". Restrictions are an example of *cotensors* , a kind of indexed limits, and their definition is as follows:

Definition 8 Let **X** be a **B**-category, g and f objects of **B**, x an **X**- object over g and h: f $\rightarrow$ g a map in **B**. A *restriction* x_h of x along h is an object over f such that $\mathbf{X}(x,y).h \equiv \mathbf{X}(x_h,y.)$

In our example x is a copy of the pc-function g and x_h is a copy of its prefix f, since h is a map. Further, x_h is totally glued to x, so it is an actual restriction: in fact, $\mathbf{X}(x_h,x) \equiv \mathbf{X}(x,x)\wedge h = h=f$. Hence, the **Ctrees** objects with restriction are precisely Cardelli's non-deterministic agents. Further, using Betti -Walters' construction, we can associate to each tree in **CTrees** its *free category with restrictions*. Once more, it is a way of passing from a presentation to a sheaf.
We would like to be able to talk of Cauchy-completeness and Cauchy-completion of objects of **CTrees** , i.e., **CA**-categories: but **C A** fails to be locally cocomplete. Hence, let us denote by **D(CA)** the distributive completion of **CA**, i.e., its cocompletion by local colimits. Then each **CA**-category can be viewed as a **D(CA)**-category and it is possible to show that Cauchy-complete **D(CA)** -categories coincide with **D(CA)** -categories with restrictions.
So, either the left operand is already a category with restrictions or it must be completed to its free category with restrictions before applying the tensor product.
Back to the case of non-deterministic left prefixing, the free category with restrictions generated by $\langle\{p\},e,\alpha\rangle$ is a tree with an *infinite* number of paths.A copy of the second tree to be concatenated is attached at the end of each of them . This corresponds to the fact that the duration of the action a in a(t):- may be *any* positive real number smaller than t. Notice that in our approach the definition of a(t):- has been given in terms of a[t]:-, in the sense that the path labelled by the constant function $f_a(t)$ in the tree modelling a[t]:- has been replaced by a *set of paths* one for each real value x in [0,t) so that the path corresponding to x is labelled by the constant functions $f_a(x)$. The agreement between the paths with labels $f_a(x)$ and $f_a(y)$, for x and y in [0,t) and x$\leq$y, is the path with label $f_a(x)$.
Let us recall that in Cardelli's account the operational behaviour of the agent a(t):p is obtained from the one of a[t]:p by adding all 'shorter sub-behaviours' corresponding to the agent a[x]:p for 0<x<t, and then the axiom p+p=p is used to prevent the multiplicity of common sub-behaviours. We would like to stress that in our framework we have followed the same approach, but the notion of agreement *prevents the multiplicity of common sub-behaviours*, (restrictions) and the continuous tree corresponding to a(t):p is indeed a standard continuous tree with the property of having infinitely many paths.

Thus, the problem of giving an explicit axiomatization of the behaviour of a(t):p w.r.t. the synchronization functor (see the non equational laws FT1-FT4 in [Car82]) is avoided.

Now we want to define the synchronization of continuous trees, which models the parallel composition (or synchronization) of computations, in analogy with the discrete case, described in [KLP87] and [KaL90], where we "lifted" a synchronization algebra [Win84] over the alphabet of actions A to the level of trees..
Let us recall that given two *strict* monoidal categories $(\underline{\mathbf{M}}_1,\otimes_1,I_1)$ and $(\underline{\mathbf{M}}_2,\otimes_2,I_2)$, a functor F from $\underline{\mathbf{M}}_1$ to $\underline{\mathbf{M}}_2$ is said to be *monoidal* iff i) there exists a morphism from I_2 to $F(I_1)$ and ii) there exists a natural transformation τ which assigns to each pair of objects A and B in $\underline{\mathbf{M}}_1$ a morphism from $F(A)\otimes_2F(B)$ to $F(A\otimes_1B)$ and τ well-behaves (see [EiK65, p.473]) w.r.t. the associativity and the identity of the tensor products.
We now define a synchronization *functor* for continuous trees.

Proposition 5 A partial synchronization operation $*$: $A\times A \rightarrow A$ lifts to a monoidal functor $[*]$: **CTrees**×**CTrees** → **CTrees**. ¶

The proof of the proposition is very similar to the one for the discrete time case, that can be found in [KLP90].
Let us also mention that we can define a *semantics* for the functor $[*]$ in analogy with what we have done in the discrete-time case in [KLP87] and [KaL90], and in particular, we are able to model Cardelli's rule ($\times \rightarrow$), at the enriched level.
Finally, the boundedness of the interval [0,t) may be relaxed in order to model Cardelli's agents with actions of *indefinite* duration (see [Car82], p.103). The agent which performs the action a for an indefinite duration is obtained by using the constant function $f_\alpha(\infty)$ and by generating the free category with restrictions.

We conclude this section by mentioning very briefly some recent work done on timed processes. Various versions of timed CSP are available, with interesting features. Let us mention for instance [Sch91], where an operational semantics in terms of a transition system is introduced and [DJS90] , where a careful analysis of relevant features like safety and liveness is carried out. A. Jeffrey is interested in process algebras in general (see [Jef91] for an interesting survey) and Moller and Tofts [MoT89] address specifically a timed version of CCS. Many of these papers assume, differing from Cardelli or Murphy [Mur91], that events are instantaneous, but with a delay. This is not a mathematical problem, in our framework, but it deserves a further discussion, likewise other stands taken on general matters. For instance, some authors view time as a partial order: unless compelling, objective reasons force us to change our minds, we prefer to assume time to be totally ordered and to ascribe to the specificity of the (true concurrent) observer the non-sequentiality of observations.
At a more technical level, we do not see relevant difficulties in fitting in our framework many of the interesting features of the papers mentioned above: just as an example, let us quote the safety problem in [DJS90] and the "time out", in particular as in [Jef91].

§ 5 Some remarks on possible timings of Mazurkiewicz Traces

In this last section we try to extend some aspects of the theory of continuous time agents in order to introduce and discuss some possible semantics of timed traces. The content of this section is tentative: we are very well aware of the naivety and the limitations of the models proposed . They aim to be just a starting point for future research. However, they allow some obvious but interesting considerations on the relationship between different timings and the kind of concurrency involved.

Let us briefly recall some terminology and a few definitions.

A *concurrent alphabet* is a pair (Σ,θ), where:

$\Sigma = \{\sigma_1,\sigma_2,...,\sigma_n\}$ is a finite alphabet;

$\theta \subseteq \Sigma \times \Sigma$ is a symmetric and irreflexive relation, called the *independency* or *concurrency relation.*

The *dependency relation* is just $\bar{\theta} = \Sigma x\Sigma - \theta$.

Let $<\Sigma,\theta>$ be a concurrent alphabet, and let us denote by $\equiv_\theta$ the transitive and reflexive closure of the relation $\rho_\theta=\{(uabv,ubav) \mid u,v\in\Sigma^*$ and $a\theta b\}$. $\equiv_\theta$ is a congruence on Σ^* and the *free partially commutative monoid* (fpcm, for short) generated by $<\Sigma,\theta>$ is the quotient structure $F(\Sigma,\theta) = \Sigma^*/\equiv_\theta$.

A *trace* on a concurrent alphabet $<\Sigma,\theta>$ is any element $\underline{t}\in F(\Sigma,\theta)$; a *trace language* is any subset $T\subseteq F(\Sigma,\theta)$.

Let us recall some basic facts about traces.

Prefix ordering on traces is defined as usual: $\underline{t} \leq \underline{t}'$ iff $\exists$ s in $F(\Sigma,\theta)$ with $\underline{t} \,.\, \underline{s} = \underline{t}'$.

Every string $v\in\Sigma^*$ can be mapped to the corresponding trace $\underline{t} = [v]_\theta$ by the *canonical homomorphism* $[_]_\theta$: $\Sigma^* \rightarrow F(\Sigma,\theta)$. A trace can be represented as a partially ordered set of symbol occurrences. It is possible to define *normal forms*. In fact, let Γ be an alphabet isomorphic with the set $C = \{A\subseteq\Sigma \mid a,b\in A \wedge a\neq b \Rightarrow a\theta b\}$ of *cliques* of θ. We can map in a unique way any trace $\underline{t} \in F(\Sigma,\theta)$ to a string $\phi(t) = \gamma_1\gamma_2 \ldots \gamma_n$, where each $\gamma_i\in\Gamma$ is (the isomorphic image of) a non empty clique of θ such that:

$$\forall\sigma\in\gamma_i \text{ there is a } \sigma'\in\gamma_{i-1} \text{ such that } \sigma\bar{\theta}\sigma' \qquad (*)$$

ϕ: $F(\Sigma,\theta) \rightarrow \Gamma^*$ is called the *normal form function.*

The normal form function allows us to define the *Foata normal form* (F.n.f.) of the trace $\underline{t}$ as the string $\phi(t)\in\Gamma^*$ .The F.n.f. allows to represent a trace in a way that expresses its maximum amount of parallelism; in fact, any symbol in the F.n.f. represents the largest set of actions that can be executed in a unique step by a set of different processors working in parallel.

If we now impose a total ordering $\sigma_1 < \sigma_2 < ... < \sigma_n$ on Σ, we can define a *backward function* ψ: $\Gamma^* \rightarrow \Sigma^*$ which associates to every string on Γ the string on Σ obtained by replacing each symbol γ_i with the sequence of elements of the corresponding clique, ordered by $<$. The set of *canonical representatives* in Σ^* of traces, which is the image of $F(\Sigma,\theta)$ through the function

$\psi \cdot \phi$, will be denoted by $CR(\Sigma,\theta)$. Furthermore, it is easy to see that $[_]_\theta \cdot \psi \cdot \phi$ is the identity function on $F(\Sigma,\theta)$.
It has been observed in [KLP90] that $F(\Sigma,\theta)$ is a meet-semilattice with respect to the obvious prefix ordering and hence an sl-2-category of *trace observers*.

In order to define timed traces, we need a *time object*.$\mathbb{T}$. The choice is quite relevant: to assume the natural numbers would lead us to ordinary traces, while the nonnegative reals would lead us to a dense timing where the analogue of the strings in Σ^* are the partial pc-functions considered in section 3. A bounded interval of the real line might have been a reasonable choice . In fact, one could argue that the independence relation θ given a priori is slightly unrealistic: if independent processes are not supposed to share resources, it is very well possible that after a certain time this condition fails to be true. The two processes might enter in a conflict situation on the scarce resources available.
A first possible approach consists in taking as concurrent alphabet $(\Sigma \times \mathbb{T},\ \theta \times \tau)$, where τ is a relation on $\mathbb{T}$. Let us assume for the moment that τ is the identity relation I on $\mathbb{T}$.
Hence, $(a,t)\theta \times I\ (b,t')$ iff $a\ \theta\ b$ and $t\ I\ t'$, i.e., $t=t'$. The free partially commutative monoid of traces generated by $(\Sigma \times \mathbb{T},\ \theta \times I)$ is defined exactly as in the ordinary case, yielding in fact standard traces on a "timed" alphabet. A trace, viewed as an equivalence class, is a set of partial pc-functions: recall that any partial pc-function can be represented as concatenation of partial *constant* functions, i.e. timed actions. Two equivalent partial pc-functions behave pretty much like words in Σ^*, and differ only for the order of permutable pairs of timed actions. They have the same overall duration and in particular the durations of independent pairs, say (a,t) and (b,t), in the same partial pc-function get added: so we get in fact an *interleaving model*. From the point of view of true concurrency this feature is rather unnatural: independent actions are not supposed to take place one after the other. On the other hand, it is interesting that the most naive and immediate extension to timed processes *within* trace theory leads directly to an interleaving model.
It is trivial to prove that the traces defined above are a meet-semilattice and hence a 2-category. Categories enriched on it, i.e., trees of timed traces look like "timed event structures", according to the theorem proved in [KLP90] for the classical case. The interest of such a strict interleaving model in approaching timed true concurrency is certainly debatable, but the model allows to get reasonable and well behaved tensor products, synchronizations and "observational" equivalences of various kind.

A slight refinement of the previous interleaving model, still in terms of standard traces, can be obtained by releasing the condition that duration of every occurrence is fixed. It could be more interesting to consider timed traces where individual durations are fixed for non-permutable atomic actions only, while an overall duration is assigned to each pair of independent timed actions. Let us consider a concurrent alphabet $(\Sigma \times \mathbb{T},\ \theta \times \tau_a)$, where τ_a is a family of relations on $\mathbb{T}$ depending on Σ, such that $(a,t)\theta \times \tau_a\ (b,t')$ iff $a\ \theta\ b$ and $t+t' = s([a]_\theta)$, that is a fixed duration depending on the trace $[a]_\theta$ in $F(\Sigma,\theta)$. Again, it is trivial to define the (standard) trace monoid and to check that we are given a meet-semilattice and

hence a 2-category. Two equivalent partial pc-functions f and g now differ in the order of permutable pairs of timed actions, i.e., such that $|f^{-1}(a)|+|f^{-1}(b)| = |g^{-1}(b)|+|g^{-1}(a)|$, where $|f^{1}(a)|$ is the duration of the action a. Also in this model enriched categories represent "timed event structures", of course in a different way. Once more, it is possible to get well behaved tensor products, synchronizations and "observational" equivalences .

Now, let us take a *non* interleaving approach, at the cost of dropping the standard trace setting. Let us start once again with the concurrent alphabet (Σ,θ) and define the trace monoid as usual. Now, let us assign a duration to each string in Σ^*, provided that strings belonging to the same trace have the same duration. Let us consider the Foata normal form of each trace, and let us give a duration to each $\gamma_i \in \Gamma$, namely the (common) duration of any $\sigma \in \gamma_i$. This provides a duration for traces, not any longer additive on independent actions, that is certainly closer to the "true concurrent" intuition. The investigation of the categorical properties of such a structure is object of future work.

Let us conclude this section with a further possible development of the previous model. If we drop the requirement that strings belonging to the same trace have the same duration, we can assign to each clique of the Foata normal form the duration which is *maximal* among the actions of the clique.

REFERENCES

[Bén65] Bénabou, J.: "Catégories relatives", C.R.A.S. 260 (1965), 3824-3827.
[Bén67] Bénabou, J.: "Introduction to Bicategories", in LNM 47, Springer Verlag, Berlin (1967), 1-77.
[Bén84] Bénabou, J.: "Wandering and wondering about the high trees", (1984), (manuscript).
[Ben89] Benson,D.:"Bialgebras: Some Foundations for Distributed and Concurrent Computation", Fundamenta Informaticae, 12 (1989), 427-486.
[BeK85] Bergstra, J.A. and Klop, J.W.: "Algebra of Communicating Processes with Abstraction". Theoretical Computer Science, Vol.37, No.1 (1985), 77-121.
[Bet80] Betti, R.: "Automi e categorie chiuse" Boll. Unione Matem. Ital. (5) 17-13 (1980), 44-58.
[BeW85] Betti, R. and Walters R.F.C. "Closed bicategories and variable category theory", Quaderno 5/P (1985), Dip. Mat.Univ. Milano.
[BeW87] Betti, R. and Walters R.F.C. "On completeness of locally internal categories", Journal of Pure and Appl. Algebra (1987), 105-117.
[BHR84] Brookes, S. D., Hoare, C.A.R., and Roscoe, A.W.: "A Theory of Communicating Sequential Processes" J.A.C.M. 31, 3 (1984), 560-599.
[Car82] Cardelli, L.: "Real Time Agents" Proc. ICALP 1982, LNCS n. 140 (1982), 94-106.
[CaW87] Carboni, A. and Walters, R.F.C.: "Cartesian Bicategories I" J. Pure Applied Algebra 49, (1987), 11-32.

[CCM90] R.Casley, R.Crew, J.Meseguer, V.Pratt, "Temporal structures", LNCS 389,1989.
[CKW87] Carboni A., Kasangian S., Walters R., "An Axiomatics for Bicategories of Modules", Journal of Pure and Appl.Algebra, 45 (1987), 127-141.
[DJS90] Davies, J., Jackson,D., Schneider, S.: "Making things happen in timed CSP" Tech. Report, Oxford, 1990.
[DMM89] Degano, P., Meseguer, J., and Montanari, U.: "Axiomatizing Net Computations and Processes", Fourth Annual Symposium on Logic in Computer Science, IEEE Computer Society Press, (1989), 175-185.
[DeM87] Degano, P. and Montanari, U.: "Concurrent Histories: A Basis for Observing Distributed Systems", Journal of Computer and System Science, 34 (1987), 422-461.
[DeL91] De Nicola, R. and Labella, A. "A functorial assessment for bisimulation" preprint, 1991
[EiK65] Eilenberg, S. and Kelly, G.M.: "Closed Categories" Proc. of the Conference on Categorical Algebra. La Jolla 1965, Springer Verlag (1966), 421-562.
[GeB87] Gerth,R., Boucher,A.:"A timed failures model for extended communicating processes", ICALP '87, Springer LNCS 267, 1987
[Hen88] Hennessy,M. *Algebraic Theory of Processes,* MIT Press, 1988.
[Jef91] Jeffrey,A.: "Timed process algebra ≠ time x process algebra", Preprint, Chalmers, 1991.
[JoG85] Josephs M. and Goswami A., "Relating computation and time" Tech. Report RR138, Department of Computer Science, University of Warwick, 1985.
[KKR83] Kasangian, S., Kelly, G.M., and Rossi, F.: "Cofibrations and the Realization of Nondeterministic Automata" Cahiers de Topologie et Géométrie Différentielle, XXIV 1 (1983), 23-46.
[KaL90] Kasangian, S. and Labella, A.:"Enriched Categorical Semantics for Distributed Calculi" to appear in Journal of Pure and Applied Algebra, 1990.
[KLP87] Kasangian, S., Labella, A. and Pettorossi, A.: "Enriched Categories for Local and Interaction Calculi", LNCS 283, Springer Verlag, Berlin (1987), 57-70.
[KaP] Kasangian, S., Pacco, W.: "An enriched categorical characterization of SCCS", preprint, 1991.
[KaV91] Kapranov, M.and Voevodsky V. :"Braided monoidal 2-categories, 2-vector spaces and Zamolodchikov's tetrahedra equations", Preprint 1991.
[Kel82] Kelly, G.M.: "Basic Concepts of Enriched Category Theory", Cambridge University Press, Cambridge (1982).
[Law73] Lawvere, F.W.: "Metric Spaces, Generalized Logic, and Closed Categories" Rendiconti del Seminario Matematico e Fisico, Milano 43 (1973), 135-166.
[Maz87] Mazurkiewicz, A.: "Trace theory" , LNCS 255, Springer Verlag, Berlin (1987), 279-324.
[MeM88] Meseguer, J., and Montanari, U.:"Petri Nets are Monoids: A New Algebraic Foundation for Net Theory" Proc. 3rd Annual Symposium on Logic in Computer Science, June 5-8, 1988, Edinburgh (Scotland), 155-164.
[Mil83] Milner, R.: "Calculi for Synchrony and Asynchrony" Theoretical Computer Science, Vol.25 (1983), 267-310.

[MoT89] Moller, F.,Tofts, C. :"A temporal calculus of communicating systems", Preprint, Edinburgh, 1989.
[Mur91] Murphy D.: "3 papers on classical concurrency theory", CS Report, Glasgow, 1991.
[Pra86] Pratt, V.: "Modelling Concurrency with Partial Orders" International Journal of Parallel Programming, v.15, 1(1986), 33-71.
[Rei85] Reisig, W.:"Petri Nets: an Introduction" EATCS Monographs on Theoretical Computer Science, Vol.4, Springer Verlag (1985).
[Sch91] Schneider, S. :"An operational semantics for timed CSP", Tech. Report, Oxford, 1991.
[Sta89] Stark,E.: "Concurrent Transition Systems", Theoretical Computer Science 64 (1989) 221-269.
[Tof88] Tofts,C.: "Temporal Ordering for Concurrency", Univ.of Edinburgh Report LFCS-49-88, 1988
[Wal81] Walters, R.F.C.:"Sheaves and Cauchy-Complete Categories" Cahiers de Top. et Géom. Diff. 22. (1981), 283-286.
[Wal82] Walters R.F.C.: "Sheaves on sites as Cauchy-complete Categories", Journal of Pure and Appl.Algebra, 24,1982, 95-102.
[Win84] Winskel, G.: "Synchronization Trees" Theoretical Computer Science, 34 (1984), 33-82.
[Win87] Winskel, G.: "Event structures", LNCS 255, Springer Verlag (1987), 325-392.