Math. Struct. in Comp. Science (19??), *vol.* ??, *pp.* ??-??. Printed in the United Kingdom

Symmetric monoidal and cartesian double categories as a semantic framework for tile logic†

ROBERTO BRUNI[1], JOSÉ MESEGUER[2] and UGO MONTANARI[1]

[1] *Dipartimento di Informatica, Università di Pisa, Italia.*
[2] *Computer Science Laboratory, SRI International, Menlo Park, CA, U.S.A.*

Received 15 January 2001

Tile systems offer a general paradigm for modular descriptions of concurrent systems, based on a set of conditional rewriting rules with side-effects. *Monoidal double categories* are a natural semantic framework for tile systems, because the mathematical structures describing system states and synchronizing actions (respectively called *configurations* and *observations* according to our terminology) are monoidal categories having the same objects (the *interfaces* of the system). In particular, configurations and observations based on net-process-like and term structures are usually described in terms of *symmetric monoidal* and *cartesian categories*, where the auxiliary structures for the rearrangement of interfaces correspond to suitable natural transformations. In this paper we discuss the lifting of these auxiliary structures to double categories. We notice that the internal construction of double categories produces a pathological asymmetric notion of natural transformation, which is fully exploited in one dimension only (e.g., for configurations or for observations, but not for both). Following Ehresmann (1963), we overcome this biased definition, introducing the notion of *generalized natural transformation* between *four* double functors (rather than two). As a consequence, the concepts of *symmetric monoidal* and *cartesian (with consistently chosen products)* double categories arise in a natural way from the corresponding ordinary versions, relating very well the auxiliary structures of configurations and observations. Moreover, the *Kelly-MacLane coherence axioms* can be lifted to our setting without effort, thanks to the characterization of two suitable *diagonal categories* which are always present in a double category. Then, symmetric monoidal and cartesian double categories are shown to offer an adequate semantic setting for *process* and *term* tile systems.

† Research supported by Office of Naval Research Contracts N00014-95-C-0225 and N00014-96-C-0114, by National Science Foundation Grant CCR-9633363, and by the Information Technology Promotion Agency, Japan, as part of the Industrial Science and Technology Frontier Program "New Models for Software Architechture" sponsored by NEDO (New Energy and Industrial Technology Development Organization). Also research supported in part by U.S. Army contract DABT63-96-C-0096 (DARPA); CNR Integrated Project *Metodi e Strumenti per la Progettazione e la Verifica di Sistemi Eterogenei Connessi mediante Reti di Comunicazione*; and Esprit Working Groups *CONFER2* and *COORDINA*. Research carried out in part while the first and the third authors were visiting at Computer Science Laboratory, SRI International, and the third author was visiting scholar at Stanford University.

Contents

1. Introduction

In this paper we characterize suitable semantic frameworks for two interesting classes of tile systems, where the similarities between the mathematical structures employed to model system states and their observable evolutions are exploited systematically.

Tile systems (Gadducci 1996; Gadducci and Montanari 2000) yield a formalism for modular descriptions of concurrent systems, extending both the SOS approach (Plotkin 1981) and context systems (Larsen and Xinxin 1990) to a framework where the rules, called *tiles*, have a very general format. Tiles define the behaviour of certain *open* (e.g., partially specified) modules, which may interact through their interfaces. Then, the behaviour of a complex system is defined as a coordinated evolution of its sub-modules. A tile has the form

$$\begin{tikzcd} \circ \arrow[r, "s"] \arrow[d, "a"'] \arrow[dr, phantom, "A"] & \circ \arrow[d, "b"] \\ \circ \arrow[r, "s'"'] & \circ \end{tikzcd}$$

also written $A : s \xrightarrow[b]{a} s'$, stating that the *initial configuration* s of the system can evolve to the *final configuration* s' producing an observable *effect* b, provided that the subcomponents of s evolve to the subcomponents of s', producing the effect a, which is the *trigger* for the application of the tile A.

The vertices of the tile are called interfaces (i.e., each configuration has both an *input* and an *output* interface). Configurations model system states. Both trigger and effect are called *observations* and model the interaction with the environment during a computation.

Tiles can be composed horizontally (synchronizing the effect of the first rule with the trigger of the second rule), vertically (computational evolutions of a certain component), and in parallel (concurrent steps) to generate larger steps. These three operations are denoted by $\_ * \_$, $\_ \cdot \_$ and $\_ \otimes \_$ respectively, and satisfy particular composition properties.

Although the tile model can be equipped with a purely logical presentation (by analogy with *rewriting logic* (Meseguer 1990; Meseguer 1992)), where the tiles are considered as special sequents subject to certain inference rules, tile systems can be more generally seen as monoidal double categories where the tiles are just cells, the configurations are arrows of the horizontal 1-category, and the side-effects are arrows of the vertical 1-category, objects being just variables which are used to connect the "static" horizontal category with the dynamic vertical evolution. It follows that *monoidal double categories* are suitable semantic models of tile systems, where the three notions of composition between tiles are represented in a natural and intuitive way.

A minimal requirement for modeling a system with tiles is that the configurations and the observations that one wants to represent must define two suitable monoidal categories, having the same objects. However, in many cases, the mathematical structures employed are very similar, in the sense that they are generated starting from (different) given sets of basic constructors, adding the same *auxiliary structure*. For example, Ferrari and Montanari (1997) have defined interesting tile models for the (asynchronous) π-calculus (Milner, Parrow and Walker 1992) and for the representation of both the operational and the abstract semantics of CCS with locations (Boudol, Castellani, Hennessy and Kiehn 1993) where both configurations and observations are *term graphs* (Eekele, Plasmeijer, and Sleep 1993). Term graphs are a reference-oriented generalization of the ordinary (value-oriented) notion of term, where the sharing of subterms can be specified also for closed (i.e., without variables) term graphs. (Terms can share variables, but shared subterms of closed terms can be freely copied, always yielding an equivalent term). They are very useful to algebraically model the notion of *name sharing* that is essential for several applications (we can think of names as links to communication channels, or to objects, or to locations, or to remote shared resources, or, also, to some *cause* in the event history of the system). We can also think of term graphs as suitable net-process-like structures (enriched with a sharing mechanism), i.e., as arrows of suitable *symmetric monoidal categories*.

Also ordinary term structures can be used to describe both configurations and observations; in this case they can be modeled by *cartesian categories* (*with chosen products*) rather than by just (symmetric) monoidal categories. In both cases the additional auxiliary structures required can be characterized in terms of suitable natural transformations, whose components are called *symmetries* (for the symmetric monoidal case), and *symmetries*, *duplicators*, and *dischargers* (for the cartesian case). Consequently, the models of such tile systems must be monoidal double categories having some additional structure. Such structure must relate the auxiliary structures of configurations and observations and must satisfy some coherence properties.

Attempts based on internal category constructions (that is, based on models of the

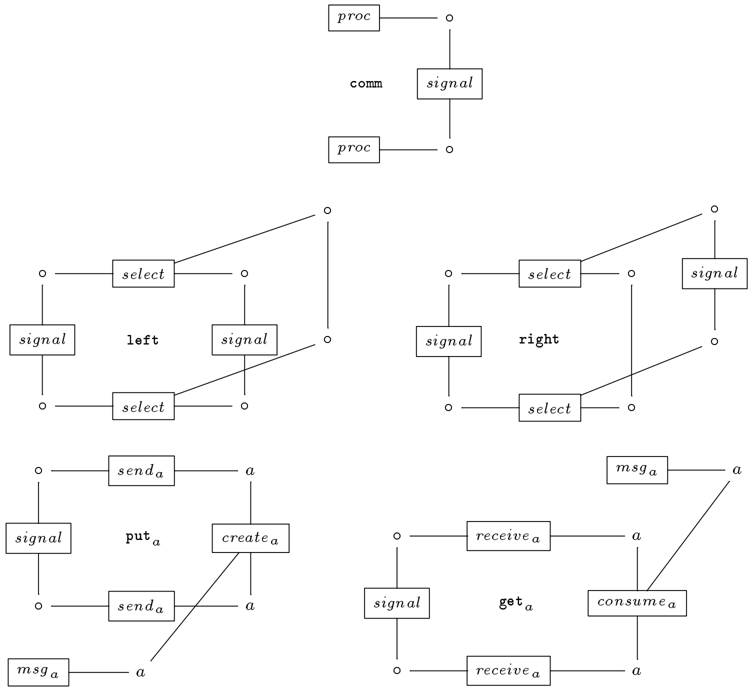


Fig. 1. A graphical presentation of the tile system $\mathcal{R}_{SAMP}$.

sketch of categories in suitable structured categories) for the definition of symmetric, and, more generally, of cartesian double categories can only lead to asymmetric models, where the auxiliary structure is fully exploited in one dimension only. We believe that this should not be the case, both conceptually and as needed by the kind of applications that we are developing; therefore, we propose in this paper a new notion of symmetric monoidal and cartesian double category that behaves in the same way in both the horizontal and vertical dimension, and that fits very well our concurrency applications.

1.1. *A Message Passing Example*

To illustrate with a simple example the nature of the auxiliary structures that we are investigating, let us consider the tile system $\mathcal{R}_{SAMP}$, whose basic tiles are depicted in Figure 1 using the "wire and box" notation. In the wire and box notation labeled boxes represent the operators and wires represent the connections between the various components, the sequential composition $\_;\_$ merges the target interface of the first argument and the source interface of the second argument, and the parallel composition $\_\otimes\_$ is represented picturing its arguments side-by-side. To improve readability, we shall sometimes draw observations using dotted wires and dotted boxes.

$\mathcal{R}_{SAMP}$ defines a computational model for *Simple Asynchronous Message Passing* systems, yielding an abstract observation of the communication flows. The basic idea is that each interface represent collections of communicating subsystems, and the observation give causality information about occurred comunications.

Let Γ be a given set of communication channels, ranged over by a. A *SAMP process* is a collection of *ports* dedicated to sending or receiving messages on some channels. We consider as interfaces the elements of $(\Gamma \cup \{\circ\})^*$, i.e., the "type" strings on the set of communication channels plus a special symbol "$\circ$" for those processes that are not yet associated with some port. We use λ to denote the empty string.

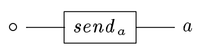
The configurations are the arrows of the free symmetric strict monoidal category freely generated by the basic constructors:

— $\mathit{proc} : \lambda \to \circ$ (basic empty process);
— $\mathit{msg}_a : \lambda \to a$ (generic message on the channel a);
— $\mathit{send}_a : \circ \to a$ (representing a port dedicated to sending messages on the channel a);
— $\mathit{receive}_a : \circ \to a$ (representing a port dedicated to receiving messages that have been posted on channel a); and
— $\mathit{select} : \circ \to \circ \otimes \circ$ (representing a binary mutual exclusion operator for the selection of i/o ports).

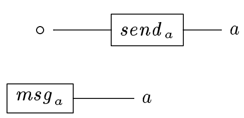
The observations are the arrows of the symmetric strict monoidal category freely generated by the basic constructors:

— $\mathit{signal} : \circ \to \circ$ (the capability to communicate, either sending or receiving);
— $\mathit{create}_a : a \to a \otimes a$ (posting of a message on channel a); and
— $\mathit{consume}_a : a \otimes a \to a$ (retrieving a produced message from channel a).

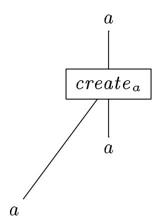
Graphically, the symmetric strict monoidal category structure of configurations and observations can be equivalently represented using acyclic wire and box diagrams where each node has at most one entering wire and at most one exiting wire (to avoid confusion, we remark that "boxes" are not nodes). Due to the parallel composition operation, some of the wire and box representations of the tiles defining $\mathcal{R}_{SAMP}$ require a three-dimensional interpretation of the pictures. For example the the tile $\mathtt{put}_a$ rewrites the initial configuration send_a below:



into the final configuration $\mathit{msg}_a \otimes \mathit{send}_a$:



producing the effect create_a:



$$\texttt{comm} : proc \xrightarrow[signal]{} proc \qquad \texttt{left} : select \xrightarrow[signal \otimes id_\circ]{signal} select \qquad \texttt{right} : select \xrightarrow[id_\circ \otimes signal]{signal} select$$

$$\texttt{put}_a : send_a \xrightarrow[create_a]{signal} msg_a \otimes send_a \qquad \texttt{get}_a : receive_a \otimes msg_a \xrightarrow[consume_a]{signal} receive_a$$

Fig. 2. The tile system $\mathcal{R}_{SAMP}$ presented using the algebraic notation.

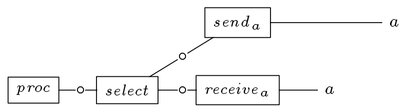
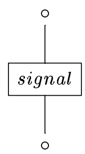


Fig. 3. The configuration $P = proc; select; (receive_a \otimes send_a)$

but only if the subcomponents can provide the capability for the communication, which is the trigger *signal*:



By adopting the algebraic notation for configurations and observations, the basic tiles of $\mathcal{R}_{SAMP}$ can be rewritten as illustrated in Figure 2.

The first tile, called `comm`, states that each basic process can decide to start a new communication, creating its own communication capability (represented by the effect *signal*).

The two tiles for the configuration *select*, labeled by `left` and `right`, implement the nondeterministic propagation of the communication capability to the ports of the process connected to the left or to the right output of the *select*.

The tile $\texttt{put}_a$ for the configuration $send_a$ states that the communication capability can be used to execute the send, creating a new message on the channel a and yielding the effect $create_a$. Analogously the tile $\texttt{get}_a$ models the reception of a message. Notice that the "cyclic" behaviour of the ports makes this example very simple and suitable for illustrating our informal presentation of process tile systems.

For example, let us consider the simple configuration

$$P = proc; select; (receive_a \otimes send_a),$$

which is in Figure 3. The rule `comm` can be applied to the box *proc* of P only if the rest of P (i.e., the configuration $select; (receive_a \otimes send_a)$) can be rewritten using as trigger the observation *signal* produced by the application of `comm`. If one tries to use the rule `left` to propagate the capability *signal* to the $receive_a$ box, then rewriting gets stuck, because no message has been produced on the channel a, and the rule $\texttt{get}_a$ cannot be

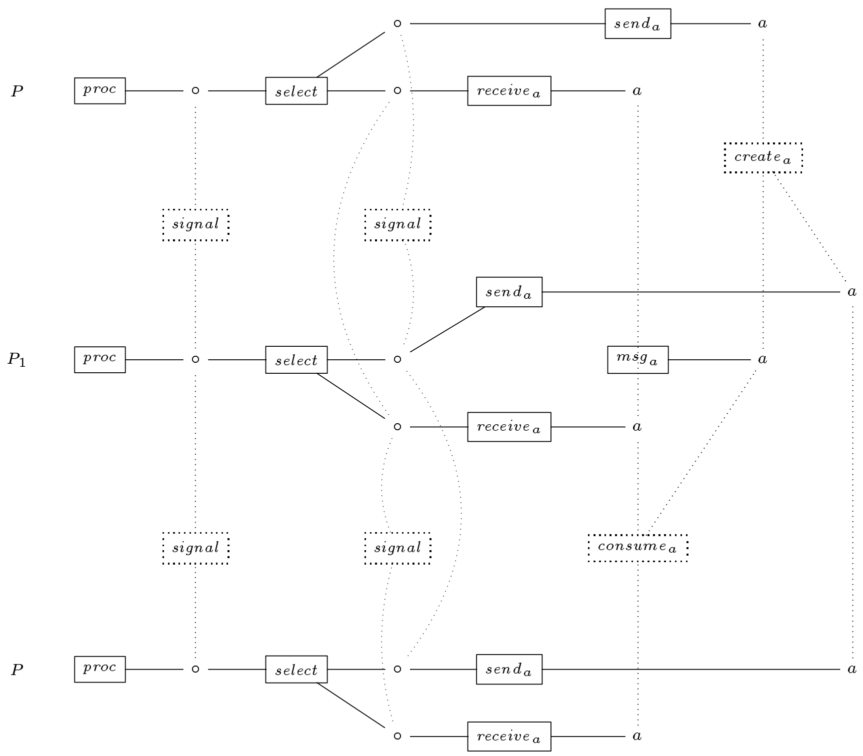


Fig. 4. The wire and box representation of a tile computation for the configuration P.

applied. However, the rule `right` can be employed to propagate the capability *signal* to the $send_a$ box. Then, the rule $\mathtt{put}_a$ can be used to conclude the transaction (in parallel with the idle rewriting of the box $receive_a$). This leads to the configuration

$$P_1 = proc; select; (receive_a \otimes msg_a \otimes send_a).$$

At the successive step, the presence of a message on the channel a enables the application of the rule $\mathtt{get}_a$ (synchronizing it with the applications of rules `comm`, `left` and the idle rewriting of the box $send_a$). The resulting computation is fully illustrated in Figure 4. (Note that the final configuration is again P).

Denoting the vertical identity of a generic configuration C by C itself, the proof of the first step of the computation is represented by the term

$$S_1 = \mathtt{comm} * \mathtt{right} * (receive_a \otimes \mathtt{put}_a)$$

in the algebra of tiles (we have used $receive_a$ to denote the vertical identity tile for the receive configuration).

Similarly, the second step of the computation is denoted by the term

$$S_2 = \mathtt{comm} * \mathtt{left} * (\mathtt{get}_a \otimes send_a).$$

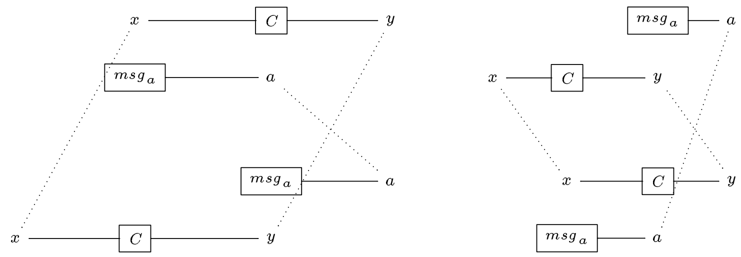


Fig. 5. The tiles for the swapping of a *message* and a generic configuration $C : x \to y$.

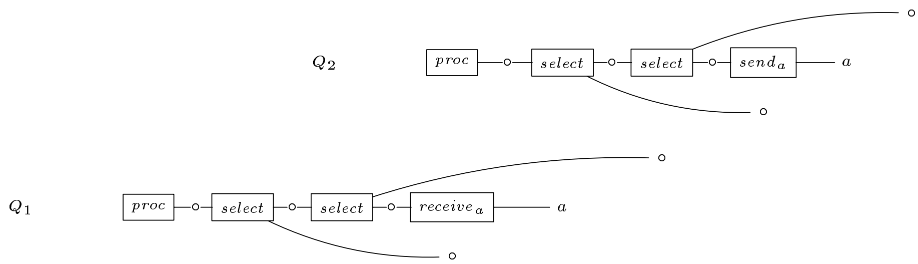


Fig. 6. The wire and box representation of the configuration $Q = Q_1 \otimes Q_2$.

The whole computation is just the vertical composition $S_1 \cdot S_2$ of the two steps.

In the wire and box notation, symmetries are represented by crossing of wires. Vertical symmetries can be used to swap configurations; for example they are necessary to put a message produced elsewhere in the system closer to an enabled $receive_a$ port, making the specification more flexible (otherwise the tile $\mathtt{put}_a$ should take into consideration the fact that the message to be consumed is not necessarily contiguous to the enabled $receive_a$).

The vertical symmetries swapping a message and a generic configuration $C : x \to y$ (i.e., any parallel and sequential composition of identities and basic constructors) with $x, y \in (\Gamma \cup \{\circ\})^{\otimes}$ (i.e., x and y are elements of the monoid freely generated by $\{\circ\} \cup \Gamma$) are graphically illustrated in Figure 5. The tile on the left states that the configuration $msg_a \otimes C$ can be rewritten into $C \otimes msg_a$ yielding the symmetry that swaps the interfaces a and y as effect. The tile on the right defines the inverse rewriting (indeed, the vertical composition of the two tiles yields the identity). Notice that the triggers of such tiles are identities, therefore they can be horizontally composed with the vertical identity of each configuration whose output interface matches the initial interface x of the configuration C. Horizontal symmetries are not needed in this case, because the vertical symmetries are expressive enough to represent all the possible communications.

For example, let us consider the configuration $Q = Q_1 \otimes Q_2$ in Figure 6, with

$$
\begin{aligned}
Q_1 &= proc; select; (id_\circ \otimes (select; (receive_a \otimes id_\circ))) \\
Q_2 &= proc; select; (id_\circ \otimes (select; (send_a \otimes id_\circ)))
\end{aligned}
$$

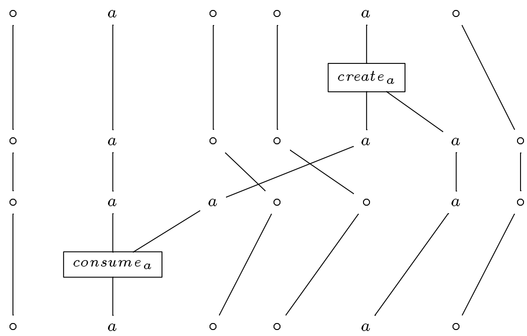


Fig. 7. The effect resulting from the sending and the retrieving of a message on the channel a in the configuration Q.

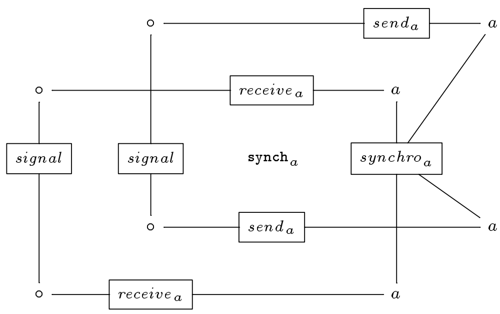


Fig. 8. The tile $\texttt{synch}_a$ for the synchronization of a send and a receive operations.

It can be easily verified that the configuration Q_2 can produce a message on the channel a (the proof is given by horizontally composing the tiles `comm`, `right`, `left` and $\texttt{put}_a$ using also some parallel composition with vertical identites when needed). However, the box $receive_a$ of Q_1 can only consume adjacent messages. The task of bringing the message closer to the receiving port can be accomplished by the vertical swapping of the message and the two wires ($id_{\circ\otimes\circ}$) that separate the message from the *receive* component. The resulting effect is graphically illustrated in Figure 7, where the "orthographic view" is preferred to the deformed "perspective view."

If we introduce an explicit synchronization mechanism in our system using the tile $\texttt{synch}_a$ represented in Figure 8, then also horizontal symmetries become necessary, as well as tiles mixing horizontal and vertical symmetries in a consistent way.

For example, the interested reader could try to synchronize the $send_a$ and the $receive_a$ operations in the configuration Q (Figure 6) without using any horizontal symmetries (it is impossible, because the *select* operator does not allow the vertical symmetries as side-effects). A solution, involving horizontal and vertical symmetries, is illustrated at the end of Section 6.

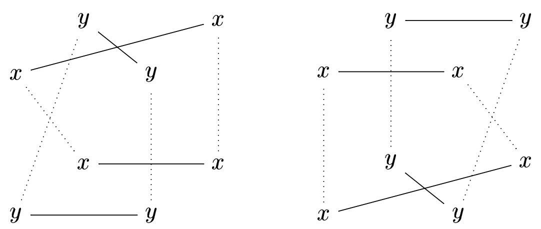


Fig. 9. Symmetries at work on both dimensions.

The idea is to combine horizontal and vertical symmetries using tiles as those pictured in Figure 9. In particular, the tile on the right in Figure 9 can be horizontally composed with any idle configuration of type $x \otimes y$ (its trigger is the identity) and perform a transformation on the spatial representation of the system, reaching a configuration where the two components of the interface are presented in reverse order. However, this creates a side-effect that must be communicated to the environment, because the protocol to interact with the transformed configuration must be adapted w.r.t. such change.

1.2. *Defining Symmetric and Cartesian Double Categories*

We already mentioned that the introduction of symmetries in both dimensions in the categorical models cannot arise from an internal construction. The first problem is that the internal definition of natural transformation is misleading: since double categories have two different notions of sequential composition, it is not clear which of them to use for a general notion of double natural transformation.

Ehresmann (1963) noticed another way of expressing natural transformations, in terms of functors toward higher fold categories. The key point is that a natural transformation is *a functorial collection of commuting squares* in the target category (also called *quartets*), i.e., a functor from a category to a suitable double category (w.r.t. one of the two possible sequential compositions). This notion can be generalized to n-fold categories by constructing the $2n$-fold category of quartets of quartets... (n times) in all the different n dimensions. Once a notion of *multiple functor* between categories of different folds has been given, then the notion of *hypertransformation* arises naturally as a multiple functor between the source n-fold category and the $2n$-fold category which is generated by the target n-fold category. This means that a hypertransformation between n-fold categories is defined on 2^n n-fold functors (we refer to the Appendix for a more detailed description of the subject).

Since double categories are 2-fold categories, this means that we need to define the 4-fold category of horizontal quartets of vertical quartets, and that the hypertransformations are defined upon $2^2 = 4$ double functors. This yields a definition of transformations which act on both dimensions, asserting the correctness of the two ways of transforming the structure (first horizontally and then vertically, or vice versa).

In the paper, we propose the equivalent definition of *generalized natural transformation* as a more concrete rephrasing of the hyper-approach for the 2-fold case. A generalized natural transformation involves four double functors, two "horizontal" natural transformations and two "vertical" natural transformations.

We can then instantiate this notion of generalized natural transformation to deal with symmetries, duplicators, and dischargers as special instances of the general notion. This study involves the complete case analysis of the possible combinations. As an interesting result, we find out that each natural transformation generates two different notions of generalized natural transformations. However, the two possible generalized symmetries are shown to be equivalent (the reason is that symmetries are isomorphisms).

The axiomatization of the basic components of the generalized natural transformations in order to state their *coherence* is a more subtle problem. The solution that we propose relies on the characterization of suitable *diagonal categories.* We start by considering two particular subclasses of cells, having either both sources or both targets equal to identities. Then, in addition to the horizontal and vertical compositions, we define two *diagonal* compositions (one for each subclass) between cells A and B such that the "upper-left" vertex of B is equal to the "lower-right" vertex of A. These operations can always be defined in a double category, and allow expressing the coherence axioms of our generalized natural transformations as the intuitive rephrasing of the well-known Kelly-MacLane coherence axioms. Moreover, the axiomatization of symmetric monoidal and cartesian double categories seems to precisely characterize concurrent tile computations.

As an important remark, we always assume that the monoidal category structure is strict. Although this assumption leads us to consider more abstract models than most concrete structures (where the associativity and the identity axioms for the monoidal operator are satisfied only up to natural isomorphisms), the notation becomes much simpler, making the whole tile framework easier to use. In the conclusion we suggest how our ideas can be extended to the non-strict case.

An alternative definition of cartesian double categories could arise by taking the notion of limits in double categories proposed by Grandis and Paré (1997). However, since the present paper investigates a much tighter notion of double products (that we have called "consistently chosen products") than the one employed by Grandis and Paré, we have preferred to reserve the analysis of the precise relationships between the two approaches to future investigations.

1.3. *Contents of the Paper*

In Section 2 we introduce the preliminary definitions regarding double categories, and introduce the notation that will be used in the rest of the paper. In particular, in Section 2.2 we formalize and explain the notion of *generalized natural transformation*, which is essential for the understanding of the concepts discussed in Sections 4 and 5.

In Section 3 we introduce the notion of *diagonal category*, that will be used in Sections 4 and 5 to simplify the formulation of coherence axioms.

In Sections 4 and 5 we incrementally enrich the basic monoidal structure of cells, first with *generalized symmetries* and then with *generalized dischargers* and *generalized duplicators*, also presenting their complete axiomatization.

Although the notion of generalized natural transformation has already appeared (with

$$a \xrightarrow{h} b \; * \; b \xrightarrow{g} c \; = \; a \xrightarrow{h*g} c \qquad\qquad a \xrightarrow{id^a} a$$

Fig. 10. Composition and identities in the horizontal 1-category.

a different formulation) in the work of Ehresmann, the material presented in these sections (i.e., the definition of symmetric strict monoidal and cartesian with consistently chosen products double categories, and the coherence axioms in particular) is, to our knowledge, original. Moreover, the mathematical structures developed in Sections 4 and 5 provide a suitable concurrent semantic framework for tile systems, as summarized in Section 6.

This work is part of our ongoing research on general mechanisms for the rewriting implementation of interesting classes of tile systems whose configurations and observations rely on common auxiliary structures. In particular, the correctness of our approach relies on the relationship between the corresponding categorical models of tile and rewriting logic. An extensive presentation of this topic can be found in (Bruni, Meseguer and Montanari 1998) (a section of that Technical Report also presents most of the results of this paper), where net-process-like and ordinary term structures are considered for both configurations and observations, yielding the notion of *process* and *term tile logic*. Other references are (Meseguer and Montanari 1998) and the Ph.D. thesis of one of the authors (Bruni 1999).

2. Double Categories and Generalized Natural Transformations

A *double category* is an internal category in **Cat**. Due to the specific structure of **Cat**, double categories admit the following presentation, adapted from Kelly and Street (1974).

Definition 1. (Double Category) A *double category* $\mathcal{D}$ consists of a collection $a, b, c, ...$ of *objects* (also called *0-cells*), a collection $h, g, f, ...$ of *horizontal arrows* (also called *horizontal 1-cells*), a collection $v, u, w, ...$ of *vertical arrows* (also called *vertical 1-cells*) and a collection $A, B, C, ...$ of *double cells* (also called *cells*).
Objects and horizontal arrows form the *horizontal 1-category* $\mathcal{H}$ (see Figure 10), with identity id^a for each object a, and composition $\_ * \_$.
Objects and vertical arrows form also a category, called the *vertical 1-category* $\mathcal{V}$ (see Figure 11), with identity id_a for each object a, and composition $\_ \cdot \_$ (sometimes we will refer to both id^a and id_a either with the object name a or with id_a, to shorten the notation, and because, when we consider lower-dimensional objects to be included in higher-dimensional ones the notions indeed coincide).
Cells are assigned *horizontal source* and *target* (which are vertical 1-cells, i.e., arrows in the vertical 1-category) and *vertical source* and *target* (which are horizontal 1-cells, i.e., arrows in the horizontal 1-category); furthermore sources and targets must be *compatible*, in the sense that, given a cell A, with vertical source h, vertical target g, horizontal source v, and horizontal target u, then h and v have the same source, g and u have the same target, the target of h is equal to the source of u, and the target of v is equal to the

$$\begin{tikzcd} a \arrow[d, "v"'] & & & \\ b & & a \arrow[d, "v \cdot u"] & a \arrow[d, "id_a"] \\ \cdot & = & c & a \\ b \arrow[d, "u"'] & & & \\ c & & & \end{tikzcd}$$

Fig. 11. Composition and identities in the vertical 1-category.

$$\begin{tikzcd} a \arrow[r, "h"] \arrow[d, "v"'] \arrow[dr, phantom, "A"] & b \arrow[d, "u"] \\ c \arrow[r, "g"'] & d \end{tikzcd}$$

Fig. 12. Graphical representation of a cell.

source of g. These constraints can be represented by the diagram in Figure 12, for which we use the notation $A : h \xrightarrow[u]{v} g$.
In addition, cells can be composed both horizontally ($\_ * \_$) and vertically ($\_ \cdot \_$) as follows: given $A : h \xrightarrow[u]{v} g$, $B : f \xrightarrow[w]{u} k$, and $C : g \xrightarrow[s]{z} h'$, then $A * B : (h * f) \xrightarrow[w]{v} (g * k)$, and $A \cdot C : h \xrightarrow[u \cdot s]{v \cdot z} h'$ are cells. Both compositions can be pictured by pasting the diagrams in Figure 13. Moreover, given a fourth cell $D : k \xrightarrow[t]{s} f'$, horizontal and vertical compositions satisfy the following *exchange law* (see also Figure 14):

$$(A \cdot C) * (B \cdot D) = (A * B) \cdot (C * D)$$

Under these rules, cells form both a horizontal category $\mathcal{D}^*$ and a vertical category $\mathcal{D}^{\cdot}$,

$$\begin{tikzcd} a \arrow[r, "h"] \arrow[d, "v"'] \arrow[dr, phantom, "A"] & b \arrow[r, "f"] \arrow[d, "u"] \arrow[dr, phantom, "B"] & b_1 \arrow[d, "w"] \\ c \arrow[r, "g"'] & d \arrow[r, "k"'] & d_1 \end{tikzcd} = \begin{tikzcd} a \arrow[r, "h * f"] \arrow[d, "v"'] \arrow[dr, phantom, "A * B"] & b_1 \arrow[d, "w"] \\ c \arrow[r, "g * k"'] & d_1 \end{tikzcd} \qquad \begin{tikzcd} a \arrow[r, "h"] \arrow[d, "v"'] \arrow[dr, phantom, "A"] & b \arrow[d, "u"] \\ c \arrow[r, "g"] \arrow[d, "z"'] \arrow[dr, phantom, "C"] & d \arrow[d, "s"] \\ c_1 \arrow[r, "h'"'] & d_2 \end{tikzcd} = \begin{tikzcd} a \arrow[r, "h"] \arrow[d, "v \cdot z"'] \arrow[dr, phantom, "A \cdot C"] & b \arrow[d, "u \cdot s"] \\ c_1 \arrow[r, "h'"'] & d_2 \end{tikzcd}$$

Fig. 13. Horizontal and vertical composition of cells.

$$\begin{tikzcd}
a \arrow[r, "h"] \arrow[dd, "v\cdot z"'] \arrow[rdd, phantom, "A\cdot C"] & b \arrow[r, "f"] \arrow[dd, "u\cdot s"] \arrow[rdd, phantom, "B\cdot D"] & b_1 \arrow[dd, "w\cdot t"] \\
 & & \\
c_1 \arrow[r, "h'"'] & d_2 \arrow[r, "f'"'] & d_3
\end{tikzcd}
=
\begin{tikzcd}
a \arrow[r, "h*f"] \arrow[d, "v"'] \arrow[rd, phantom, "A*B"] & b_1 \arrow[d, "w"] \\
c \arrow[r, "g*k"] \arrow[d, "z"'] \arrow[rd, phantom, "C*D"] & d_1 \arrow[d, "t"] \\
c_1 \arrow[r, "h'*f'"'] & d_3
\end{tikzcd}$$

Fig. 14. The exchange law.

$$\begin{tikzcd}
a \arrow[r, "h"] \arrow[d, "a"'] \arrow[rd, phantom, "1^h"] & b \arrow[r, "g"] \arrow[d, "b"] \arrow[rd, phantom, "1^g"] & c \arrow[d, "c"] \\
a \arrow[r, "h"'] & b \arrow[r, "g"'] & c
\end{tikzcd}
=
\begin{tikzcd}
a \arrow[r, "h*g"] \arrow[d, "a"'] \arrow[rd, phantom, "1^{h*g}"] & c \arrow[d, "c"] \\
a \arrow[r, "h*g"'] & c
\end{tikzcd}
\qquad
\begin{tikzcd}
a \arrow[r, "a"] \arrow[d, "v"'] \arrow[rd, phantom, "1_v"] & a \arrow[d, "v"] \\
b \arrow[r, "b"] \arrow[d, "u"'] \arrow[rd, phantom, "1_u"] & b \arrow[d, "u"] \\
c \arrow[r, "c"'] & c
\end{tikzcd}
=
\begin{tikzcd}
a \arrow[r, "a"] \arrow[d, "v\cdot u"'] \arrow[rd, phantom, "1_{v\cdot u}"] & a \arrow[d, "v\cdot u"] \\
c \arrow[r, "c"'] & c
\end{tikzcd}$$

Fig. 15. Composition of identities.

with respective identities $1_v : a \xrightarrow[v]{v} c$ and $1^h : h \xrightarrow[b]{a} h$. Given $1^h : h \xrightarrow[b]{a} h$ and $1^g : g \xrightarrow[c]{b} g$, the equation $1^h * 1^g = 1^{h*g}$ must hold (and similarly for vertical composition of horizontal identities), as illustrated in Figure 15. Furthermore, horizontal and vertical identities of identities coincide, i.e., $1_{id_a} = 1^{id^a}$ and are denoted by the simpler notation 1_a (or just a).

Example 1. (Quartet Category) Given a category $\mathcal{C}$, we denote by $\square\mathcal{C}$ the *category of quartets* of $\mathcal{C}$: its objects are the objects of $\mathcal{C}$, its horizontal and vertical arrows are the arrows of $\mathcal{C}$, and its cells are the commuting square diagrams of arrows in $\mathcal{C}$, i.e.,

$$\begin{tikzcd}
a \arrow[r, "f"] \arrow[d, "v"'] & b \arrow[d, "u"] \\
c \arrow[r, "g"'] & d
\end{tikzcd}$$

where $f, v, u, g \in \mathcal{C}$ and $f; u = v; g$ in $\mathcal{C}$. The horizontal and vertical categories of $\square\mathcal{C}$ are usually denoted by $\square\square\mathcal{C}$ and $\genfrac{}{}{0pt}{}{\square}{\square}\mathcal{C}$ respectively. Notice that $\mathcal{C}$ is both the horizontal and the vertical 1-category of $\square\mathcal{C}$.

A double category $\mathcal{D}$ has two possible interpretations as an internal category in **Cat**. In fact, thanks to the symmetric rôle played by the horizontal and vertical dimensions of a double category, it is possible to adopt a *transposed* approach in the internal construction:

$$
\text{(in } \mathcal{D}^{\cdot}\text{)} \qquad
\begin{tikzcd}
h \arrow[d, Rightarrow, "A"'] \\
g
\end{tikzcd}
\qquad
\begin{tikzcd}
s^*(h) \arrow[r, "h"] \arrow[d, dotted, "s^*(A)"'] \arrow[dr, phantom, "\overset{A}{\Rightarrow}"] & t^*(h) \arrow[d, dotted, "t^*(A)"] \\
s^*(g) \arrow[r, "g"'] & t^*(g)
\end{tikzcd}
\qquad \text{(in } \mathcal{D}^*\text{)}
$$

Fig. 16. Horizontal category of cells as internal construction.

$$
\text{(in } \mathcal{D}^*\text{)} \qquad v \overset{A}{\Longrightarrow} u \qquad
\begin{tikzcd}
s^{\cdot}(v) \arrow[r, "s^{\cdot}(A)"] \arrow[d, dotted, "v"'] \arrow[dr, phantom, "\overset{A}{\Downarrow}"] & s^{\cdot}(u) \arrow[d, dotted, "u"] \\
t^{\cdot}(v) \arrow[r, "t^{\cdot}(A)"'] & t^{\cdot}(u)
\end{tikzcd}
\qquad \text{(in } \mathcal{D}^{\cdot}\text{)}
$$

Fig. 17. Vertical category of cells as internal construction.

1 As the internal category $(\mathcal{V}, \mathcal{D}^{\cdot}, s^*, t^*, \_*\_, i^*)$, where[†] the functors s^* and t^* map each arrow onto its corresponding horizontal source and target respectively, the functor $\_*\_$ defines horizontal composition of cells, and the functor i^* maps each (vertical) arrow of $\mathcal{V}$ onto its (horizontal) identity cell. This corresponds to picturing a generic cell $A : h \to g$ of $\mathcal{D}^{\cdot}$ as shown in Figure 16.

2 As the internal category $(\mathcal{H}, \mathcal{D}^*, s^{\cdot}, t^{\cdot}, \_\cdot\_, i^{\cdot})$ where the functors $s^{\cdot}$ and $t^{\cdot}$ map each arrow of $\mathcal{D}^*$ to its corresponding horizontal source and target, the functor $\_\cdot\_$ defines vertical composition of cells according to the source and target projections $s^{\cdot}$ and $t^{\cdot}$, and $i^{\cdot}(h) = 1^h$ for each horizontal arrow $h \in \mathcal{H}$ (see Figure 17)

Given two double categories $\mathcal{D}$ and $\mathcal{E}$, a *double functor* $F : \mathcal{D} \to \mathcal{E}$ is a 4-tuple of functions mapping objects to objects, horizontal and vertical arrows to horizontal and vertical arrows, and cells to cells, preserving identities and compositions of all kinds. Notice that, since a double category is a cat-object in **Cat**, a double functor can be equivalently defined as a pair (F_0, F_1) of functors satisfying the conditions of internal functoriality. The two notions coincide, because each functor in (F_0, F_1) is a pair of mappings on objects and arrows preserving the category structure. We denote by **DCat** the category of double categories and double functors.

2.1. *Inverse*

Since double categories have two operations of composition, the definition of the inverse of a cell is not straightforward. We propose the following:

† Remember that the category $\mathcal{V}$ has objects in $\mathcal{O}$ and arrows in $\mathcal{V}$ equipped with composition $\_\cdot\_$, and that the objects of the category $\mathcal{D}^{\cdot}$ are horizontal arrows, while its morphisms are cells equipped with vertical composition $\_\cdot\_$.

$$\begin{tikzcd} a \arrow[r, "h"] \arrow[d, "v"'] \arrow[dr, phantom, "A"] & b \arrow[d, "u"] \\ c \arrow[r, "g"'] & d \end{tikzcd} \qquad \begin{tikzcd} b \arrow[r, "h^{-1}"] \arrow[d, "u"'] \arrow[dr, phantom, "A^*"] & a \arrow[d, "v"] \\ d \arrow[r, "g^{-1}"'] & c \end{tikzcd}$$

$$\begin{tikzcd} c \arrow[r, "g"] \arrow[d, "v^{-1}"'] \arrow[dr, phantom, "A^{\cdot}"] & d \arrow[d, "u^{-1}"] \\ a \arrow[r, "h"'] & b \end{tikzcd} \qquad \begin{tikzcd} d \arrow[r, "g^{-1}"] \arrow[d, "u^{-1}"'] \arrow[dr, phantom, "A^{-1}"] & c \arrow[d, "v^{-1}"] \\ b \arrow[r, "h^{-1}"'] & a \end{tikzcd}$$

Fig. 18. Generalized inverse for the cell A.

Definition 2. (Generalized Inverse) Let $A : h \xrightarrow[u]{v} g$ be a cell in a double category $\mathcal{D}$. We say that cell A has a $*$-inverse if and only if there exists a cell A^* such that $A * A^* = 1_v$, and $A^* * A = 1_u$ (i.e., A^* is the inverse of A w.r.t. the *horizontal* composition $*$, and this implies the existence of inverses of the horizontal arrows on the border of A). Similarly, the $\cdot$-inverse $A^{\cdot}$, if it exists, satisfies the equations $A \cdot A^{\cdot} = 1^h$, $A^{\cdot} \cdot A = 1^g$ (this implies the existence of inverses of the vertical arrows on the border of A).
Then, A has a *generalized inverse* if and only if A has both a $*$-inverse and a $\cdot$-inverse, and there exists a cell A^{-1} such that:
— $A^{-1} \cdot A^* = 1^{g^{-1}}$,
— $A^* \cdot A^{-1} = 1^{h^{-1}}$ (i.e., A^{-1} is the $\cdot$-inverse of A^*),
— $A^{-1} * A^{\cdot} = 1_{u^{-1}}$, and
— $A^{\cdot} * A^{-1} = 1_{v^{-1}}$ (i.e., A^{-1} is also the $*$-inverse of $A^{\cdot}$).

This definition can be summarized by saying that the $*$-inverse $(A^{\cdot})^*$ of the $\cdot$-inverse of cell A is equal to the $\cdot$-inverse $(A^*)^{\cdot}$ of the $*$-inverse of A, and it is denoted by A^{-1} (see Figure 18).

For instance, it follows that $(A \cdot A^{\cdot}) * (A^* \cdot A^{-1}) = (A * A^*) \cdot (A^{\cdot} * A^{-1}) = 1_a$, and that $(A^{-1})^{-1} = A$.

2.2. *Natural Transformations between Double Functors*

Let $F, G : \mathcal{D} \to \mathcal{E}$ be two double functors. Following the internal construction approach, an internal natural transformation is an arrow in **Cat** which satisfies the naturality conditions w.r.t. one composition and which is functorial w.r.t. the other composition. Thus, it is essential to specify what the *internal* representations of $\mathcal{D}$ and $\mathcal{E}$ are.

In (Ehresmann 1963) the notion of *hypertransformation* is presented as the generalization of natural transformations to the n-fold case (see the Appendix). We propose the following definition of *generalized natural transformation* as a more concrete rephrasing of the hyper-view for double categories (in the original approach, emphasis was given to showing that the category **MCat** of multiple functors is cartesian closed). As a matter of notation, we call natural *comp*-transformation a transformation which satisfies the naturality requirement w.r.t. the composition operator *comp* and which is functorial w.r.t. the remaining composition operator. A generalized natural transformation is the key to expressing relationships between natural $*$-transformations and natural $\cdot$-transformations (which are the two possible notions of internal natural transformation suggested by the internal category viewpoint).

Definition 3. (Generalized Natural Transformation) Let $\mathcal{D}$ and $\mathcal{E}$ be double categories. Given a 4-tuple $(F_{00}, F_{10}, F_{01}, F_{11})$ of double functors from $\mathcal{D}$ to $\mathcal{E}$, a *generalized natural transformation* is a 5-tuple $(\alpha_{0\_}, \alpha_{1\_}; \beta_{\_0}, \beta_{\_1}; \eta)$, which is pictured as the cell

$$\begin{tikzcd} F_{00} \arrow[r, "\alpha_{0\_}"] \arrow[d, "\beta_{\_0}"'] \arrow[dr, phantom, "\eta"] & F_{01} \arrow[d, "\beta_{\_1}"] \\ F_{10} \arrow[r, "\alpha_{1\_}"'] & F_{11} \end{tikzcd}$$

where:

- for $i = 0, 1$, the symbol $\alpha_{i\_}$ denotes a natural $*$-transformation from F_{i0} to F_{i1}, i.e., $\alpha_{i\_}$ is also a functor from the category $\mathcal{V}$ of vertical arrows (that is, the objects of $\mathcal{D}^*$) of $\mathcal{D}$ to the category $\mathcal{E}^{\cdot}$,
- for $i = 0, 1$, the symbol $\beta_{\_i}$ denotes a natural $\cdot$-transformation from F_{0i} to F_{1i}, i.e., $\beta_{\_i}$ defines also a functor from the category $\mathcal{H}$ of horizontal arrows (that is, the objects of $\mathcal{D}^{\cdot}$) of $\mathcal{D}$ to the category $\mathcal{E}^*$, and
- the symbol η defines both a natural transformation from $\alpha_{0\_}$ to $\alpha_{1\_}$ (seen as functors from $\mathcal{V}$ to $\mathcal{E}^{\cdot}$) and also from $\beta_{\_0}$ to $\beta_{\_1}$ (seen as functors from $\mathcal{H}$ to $\mathcal{E}^*$).

To shorten the notation, we will denote the generalized natural transformation just by η, using a diagram as the one above to represent also its other components.

We now explain more in detail the previous definition. First, consider the double functors $F_{00}, F_{01} : \mathcal{D} \to \mathcal{E}$. A natural $*$-transformation $\alpha_{0\_} : F_{00} \Rightarrow F_{01} : \mathcal{D} \to \mathcal{E}$ is a functor (i.e., an arrow in **Cat**) from the category $\mathcal{V}$ of objects of $\mathcal{D}^*$ to $\mathcal{E}^{\cdot}$, which satisfies the equations of internal natural transformations. Thus, the functor $\alpha_{0\_}$ is a natural transformation from F_{00} to F_{01} w.r.t. horizontal composition, i.e., for each cell A of $\mathcal{D}$ we have the naturality law $F_{00}A * \alpha_{0\_,u} = \alpha_{0\_,v} * F_{01}A$.

$$\begin{tikzcd} a \arrow[r, "h"] \arrow[d, "v"'] \arrow[dr, phantom, "A"] & b \arrow[d, "u"] \\ c \arrow[r, "g"'] & d \end{tikzcd} \qquad \begin{tikzcd} F_{00}a \arrow[r, "F_{00}h"] \arrow[d, "F_{00}v"'] \arrow[dr, phantom, "F_{00}A"] & F_{00}b \arrow[r, "\alpha_{0\_,b}"] \arrow[d, "F_{00}u"] \arrow[dr, phantom, "\alpha_{0\_,u}"] & F_{01}b \arrow[d, "F_{01}u"] \\ F_{00}c \arrow[r, "F_{00}g"'] & F_{00}d \arrow[r, "\alpha_{0\_,d}"'] & F_{01}d \end{tikzcd} = \begin{tikzcd} F_{00}a \arrow[r, "\alpha_{0\_,a}"] \arrow[d, "F_{00}v"'] \arrow[dr, phantom, "\alpha_{0\_,v}"] & F_{01}a \arrow[r, "F_{01}h"] \arrow[d, "F_{01}v"] \arrow[dr, phantom, "F_{01}A"] & F_{01}b \arrow[d, "F_{01}u"] \\ F_{00}c \arrow[r, "\alpha_{0\_,c}"'] & F_{01}c \arrow[r, "F_{01}g"'] & F_{01}d \end{tikzcd}$$

It also follows that, for each horizontal arrow $h : a \to b$ in $\mathcal{D}$, we have $F_{00}h * \alpha_{0\_,b} = \alpha_{0\_,a} * F_{01}h$, that is the object component of functor $\alpha_{0\_}$ defines a natural transformation between the components of F_{00} and F_{01} on the horizontal 1-category.

Consider the double functors $F_{10}, F_{11} : \mathcal{D} \to \mathcal{E}$ and a natural $*$-transformation $\alpha_{1\_}$: $F_{10} \Rightarrow F_{11} : \mathcal{D} \to \mathcal{E}$ between them. Then, $\alpha_{1\_}$ also defines a functor from $\mathcal{V}$ to $\mathcal{E}^{\cdot}$ and satisfies the naturality equation w.r.t. horizontal composition. Notice that the functors

$\alpha_{0\_}$ and $\alpha_{1\_}$ have the same source and target categories. The generalized natural transformation η acts as a natural transformation $\eta : \alpha_{0\_} \Rightarrow \alpha_{1\_} : \mathcal{V} \to \mathcal{E}^{\cdot}$ between them. Thus, the natural transformation η associates with each object a of $\mathcal{V}$ (i.e., an object of $\mathcal{D}$) an arrow η_a of $\mathcal{E}^{\cdot}$ (i.e., a cell of $\mathcal{E}$) in such a way that the equation $\alpha_{0\_,v} \cdot \eta_c = \eta_a \cdot \alpha_{1\_,v}$ holds for each arrow $v : a \to c$ in $\mathcal{V}$.

$$
\begin{tikzcd}
a \arrow[d, "v"'] \\
c
\end{tikzcd}
\qquad
\begin{tikzcd}
F_{00}a \arrow[r, "\alpha_{0\_,a}"] \arrow[d, "F_{00}v"'] \arrow[dr, phantom, "\alpha_{0\_,v}"] & F_{01}a \arrow[d, "F_{01}v"] \\
F_{00}c \arrow[r, "\alpha_{0\_,c}"] \arrow[d, "s^*(\eta_c)"'] \arrow[dr, phantom, "\eta_c"] & F_{01}c \arrow[d, "t^*(\eta_c)"] \\
F_{10}c \arrow[r, "\alpha_{1\_,c}"'] & F_{11}c
\end{tikzcd}
=
\begin{tikzcd}
F_{00}a \arrow[r, "\alpha_{0\_,a}"] \arrow[d, "s^*(\eta_a)"'] \arrow[dr, phantom, "\eta_a"] & F_{01}a \arrow[d, "t^*(\eta_a)"] \\
F_{10}a \arrow[r, "\alpha_{1\_,a}"] \arrow[d, "F_{10}v"'] \arrow[dr, phantom, "\alpha_{1\_,v}"] & F_{11}a \arrow[d, "F_{11}v"] \\
F_{10}c \arrow[r, "\alpha_{1\_,c}"'] & F_{11}c
\end{tikzcd}
$$

This implies that $F_{00}v \cdot s^*(\eta_c) = s^*(\eta_a) \cdot F_{10}v$, and $F_{01}v \cdot t^*(\eta_c) = t^*(\eta_a) \cdot F_{11}v$, i.e., $s^*(\eta)$ (respectively, $t^*(\eta)$) is a natural transformation between the projections on the vertical 1-category of functors F_{00} and F_{10} (F_{01} and F_{11}, respectively).

A similar reasoning can be applied to the orthogonal representations of $\mathcal{D}$ and $\mathcal{E}$, defining two natural $\cdot$-transformations $\beta_{\_0} : F_{00} \Rightarrow F_{10} : \mathcal{D} \to \mathcal{E}$ and $\beta_{\_1} : F_{01} \Rightarrow F_{11} : \mathcal{D} \to \mathcal{E}$, which are functors from the category $\mathcal{H}$ of objects of $\mathcal{D}^{\cdot}$ to $\mathcal{E}^*$, and satisfy the equations $F_{00}A \cdot \beta_{\_0,g} = \beta_{\_0,h} \cdot F_{10}A$ (see picture below), and $F_{01}A \cdot \beta_{\_1,g} = \beta_{\_1,h} \cdot F_{11}A$ for each cell A of $\mathcal{D}$.

$$
\begin{tikzcd}
a \arrow[r, "h"] \arrow[d, "v"'] \arrow[dr, phantom, "A"] & b \arrow[d, "u"] \\
c \arrow[r, "g"'] & d
\end{tikzcd}
\qquad
\begin{tikzcd}
F_{00}a \arrow[r, "F_{00}h"] \arrow[d, "F_{00}v"'] \arrow[dr, phantom, "F_{00}A"] & F_{00}b \arrow[d, "F_{00}u"] \\
F_{00}c \arrow[r, "F_{00}g"] \arrow[d, "\beta_{\_0,c}"'] \arrow[dr, phantom, "\beta_{\_0,g}"] & F_{00}d \arrow[d, "\beta_{\_0,d}"] \\
F_{10}c \arrow[r, "F_{10}g"'] & F_{10}d
\end{tikzcd}
=
\begin{tikzcd}
F_{00}a \arrow[r, "F_{00}h"] \arrow[d, "\beta_{\_0,a}"'] \arrow[dr, phantom, "\beta_{\_0,h}"] & F_{00}b \arrow[d, "\beta_{\_0,b}"] \\
F_{10}a \arrow[r, "F_{10}h"] \arrow[d, "F_{10}v"'] \arrow[dr, phantom, "F_{10}A"] & F_{10}b \arrow[d, "F_{10}u"] \\
F_{10}c \arrow[r, "F_{10}g"'] & F_{10}d
\end{tikzcd}
$$

The generalized natural transformation η also defines a natural transformation from $\beta_{\_0}$ to $\beta_{\_1}$. Thus, for each arrow $h : a \to b$ of $\mathcal{H}$ we get:

$$
a \xrightarrow{h} b
\qquad
\begin{tikzcd}
F_{00}a \arrow[r, "F_{00}h"] \arrow[d, "\beta_{\_0,a}"'] \arrow[dr, phantom, "\beta_{\_0,h}"] & F_{00}b \arrow[r, "s^{\cdot}(\eta_b)"] \arrow[d, "\beta_{\_0,b}"'] \arrow[dr, phantom, "\eta_b"] & F_{01}b \arrow[d, "\beta_{\_1,b}"] \\
F_{10}a \arrow[r, "F_{10}h"'] & F_{10}b \arrow[r, "t^{\cdot}(\eta_b)"'] & F_{11}b
\end{tikzcd}
=
\begin{tikzcd}
F_{00}a \arrow[r, "s^{\cdot}(\eta_a)"] \arrow[d, "\beta_{\_0,a}"'] \arrow[dr, phantom, "\eta_a"] & F_{01}a \arrow[r, "F_{01}h"] \arrow[d, "\beta_{\_1,a}"'] \arrow[dr, phantom, "\beta_{\_1,h}"] & F_{01}b \arrow[d, "\beta_{\_1,b}"] \\
F_{10}a \arrow[r, "t^{\cdot}(\eta_a)"'] & F_{11}a \arrow[r, "F_{11}h"'] & F_{11}b
\end{tikzcd}
$$

It follows that, for each object $a \in \mathcal{O}$, the shape of the cell η_a is

$$
\begin{tikzcd}
F_{00}a \arrow[r, "\alpha_{0\_,a}"] \arrow[d, "\beta_{\_0,a}"'] \arrow[dr, phantom, "\eta_a"] & F_{01}a \arrow[d, "\beta_{\_1,a}"] \\
F_{10}a \arrow[r, "\alpha_{1\_,a}"'] & F_{11}a
\end{tikzcd}
$$

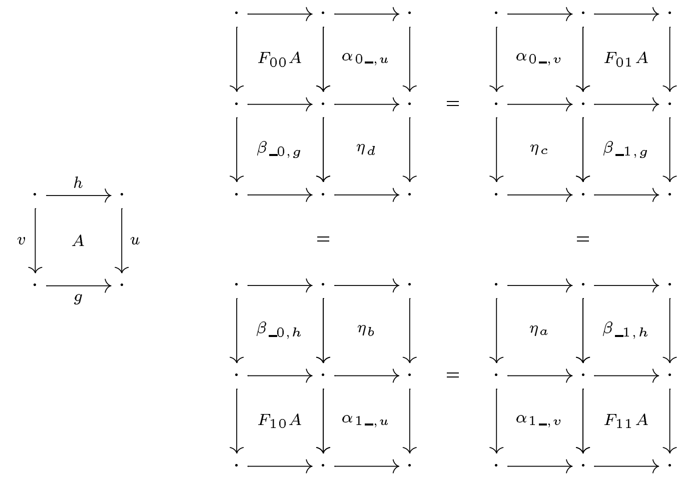


Fig. 19. Some properties of generalized natural transformations.

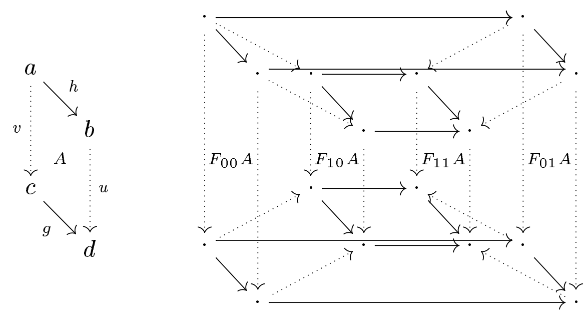


Fig. 20. The commuting hypercube.

As an example, it follows directly from the definition that, given a cell $A : h \xrightarrow[u]{v} g$, all the cell pastings in Fiugure 19 yield identical results.

All the naturality equations are faithfully represented by the commuting hypercube in Figure 20 (to make easier the interpretation, we draw vertical arrows as dotted lines):

The hypercube contains 16 vertices, 24 faces, and 8 cubes. Each vertex is the image of one of the four corner objects of cell A through one of the four functors under consideration. There are eight *empty* faces whose border involves either only vertical or only horizontal arrows. All the other 16 faces are cells of the double category $\mathcal{E}$. Four cells are the image of A w.r.t. the four different functors (see figure). Four cells are the components at h and g of the natural $*$-transformations $\alpha_{0\_}$ and $\alpha_{1\_}$. Four cells are the components at v and u of the natural $\cdot$-transformations $\beta_{\_0}$ and $\beta_{\_1}$. The remaining four cells are the components at the objects of the generalized natural transformation η. Each cube has two empty faces. The other four faces commute, in the sense that they give

$$
\begin{tikzcd}
F \arrow[r, "\alpha"] \arrow[d, "1_F"'] \arrow[dr, phantom, "1^\alpha"] & G \arrow[d, "1_G"] \\
F \arrow[r, "\alpha"'] & G
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
F \arrow[r, "1_F"] \arrow[d, "\beta"'] \arrow[dr, phantom, "1_\beta"] & F \arrow[d, "\beta"] \\
H \arrow[r, "1_H"'] & H
\end{tikzcd}
$$

Fig. 21. Natural $*$- and $\cdot$-transformations as generalized transformations.

a naturality equation. It follows that the hypercube yields eight equations for each cell A. However, the naturalities of η are both replicated for the two components of each transformation, therefore, there are six distinct equations. The functoriality axioms are given by composing the hypercubes, either one below the other, or one in front of the other.

Remark 1. Notice that the notion of generalized natural transformation generalizes natural $*$- and $\cdot$-transformations. Indeed, a natural $*$-transformation $\alpha : F \Rightarrow G : \mathcal{D} \to \mathcal{E}$ yields a corresponding generalized natural transformation 1^α. Analogously, a natural $\cdot$-transformation $\beta : F \Rightarrow H : \mathcal{D} \to \mathcal{E}$ yields a corresponding generalized natural transformation 1_β (see Figure 21).

Remark 2. Natural $*$- and $\cdot$-transformations are instances of a more general pattern. Notice that, if we consider only two generic double functors F and G, then the allowed generalized natural transformations, where the four double functors are chosen from the set $\{F, G\}$, have $2^4 = 16$ possible shapes. Only six of them do not involve transformations from G to F:

$$
\begin{tikzcd}
F \arrow[r] \arrow[d] \arrow[dr, phantom, "1"] & F \arrow[d] \\
F \arrow[r] & F
\end{tikzcd}
\quad
\begin{tikzcd}
F \arrow[r] \arrow[d] \arrow[dr, phantom, "2"] & F \arrow[d] \\
F \arrow[r] & G
\end{tikzcd}
\quad
\begin{tikzcd}
F \arrow[r] \arrow[d] \arrow[dr, phantom, "3"] & F \arrow[d] \\
G \arrow[r] & G
\end{tikzcd}
\quad
\begin{tikzcd}
F \arrow[r] \arrow[d] \arrow[dr, phantom, "4"] & G \arrow[d] \\
F \arrow[r] & G
\end{tikzcd}
\quad
\begin{tikzcd}
F \arrow[r] \arrow[d] \arrow[dr, phantom, "5"] & G \arrow[d] \\
G \arrow[r] & G
\end{tikzcd}
\quad
\begin{tikzcd}
G \arrow[r] \arrow[d] \arrow[dr, phantom, "6"] & G \arrow[d] \\
G \arrow[r] & G
\end{tikzcd}
$$

In Sections 4 and 5 we instantiate the general notion of generalized natural transformation to characterize the auxiliary structures of process and tile systems. Such correspondence is hinted at in Section 6, using the example presented in the introduction.

3. Diagonal Categories

Sometimes, due to the particular kind of cells involved in a complex composition, it is possible to adopt a more concise and convenient notation. This fact follows by observing that, for any double category, it is always possible to characterize two suitable *diagonal* sub-categories.

In fact, those cells having identities as both horizontal and vertical target are the arrows of a *diagonal* category $\mathcal{D}^\triangleleft$ whose composition $\_ \triangleleft \_$ is defined uniquely as pictured below.

$$
\begin{tikzcd}
a \arrow[r, "h"] \arrow[d, "v"'] \arrow[dr, phantom, "A"] & b \arrow[d, "b"] \\
b \arrow[r, "b"'] & b
\end{tikzcd}
\lhd
\begin{tikzcd}
b \arrow[r, "g"] \arrow[d, "u"'] \arrow[dr, phantom, "B"] & c \arrow[d, "c"] \\
c \arrow[r, "c"'] & c
\end{tikzcd}
=
\begin{tikzcd}
a \arrow[r] \arrow[d] \arrow[dr, phantom, "A"] & b \arrow[r] \arrow[d] \arrow[dr, phantom, "1^g"] & c \arrow[d] \\
b \arrow[r] \arrow[d] \arrow[dr, phantom, "1_u"] & b \arrow[r] \arrow[d] \arrow[dr, phantom, "B"] & c \arrow[d] \\
c \arrow[r] & c \arrow[r] & c
\end{tikzcd}
= (A * 1^g) \cdot (1_u * B)
$$

In a similar way, we can define a diagonal composition $\_ \rhd \_$ for those cells having identities as both horizontal and vertical source (yielding the second diagonal category $\mathcal{D}^{\rhd}$):

$$
\begin{tikzcd}
a \arrow[r, "a"] \arrow[d, "a"'] \arrow[dr, phantom, "A"] & a \arrow[d, "v"] \\
a \arrow[r, "h"'] & b
\end{tikzcd}
\rhd
\begin{tikzcd}
b \arrow[r, "b"] \arrow[d, "b"'] \arrow[dr, phantom, "B"] & b \arrow[d, "u"] \\
b \arrow[r, "g"'] & c
\end{tikzcd}
=
\begin{tikzcd}
a \arrow[r] \arrow[d] \arrow[dr, phantom, "A"] & a \arrow[r] \arrow[d] \arrow[dr, phantom, "1_v"] & a \arrow[d] \\
a \arrow[r] \arrow[d] \arrow[dr, phantom, "1^h"] & b \arrow[r] \arrow[d] \arrow[dr, phantom, "B"] & b \arrow[d] \\
b \arrow[r] & b \arrow[r] & c
\end{tikzcd}
= (A * 1_v) \cdot (1^h * B)
$$

4. Symmetric Monoidal Double Categories

As a matter of notation, in what follows we favour the horizontal dimension, by using the common symbols associated with ordinary symmetries, duplicators and dischargers to denote natural $*$-transformations rather than natural $\cdot$-transformations. However, we remark that all such notions, being based on that of generalized natural transformation, treat the horizontal and vertical dimensions equally.

Definition 4. (Strict Monoidal Double Category) A *strict monoidal double category*, sMD in the following, is a triple $(\mathcal{D}, \otimes, e)$, where:

— $\mathcal{D}$ is the underlying double category,
— $\otimes : \mathcal{D} \times \mathcal{D} \to \mathcal{D}$ is a double functor called the *tensor product*, and
— e is an object of $\mathcal{D}$ called the *unit object*,

such that the following diagrams commute:

$$
\begin{tikzcd}
\mathcal{D} \times \mathcal{D} \times \mathcal{D} \arrow[r, "\otimes \times 1"] \arrow[d, "1 \times \otimes"'] & \mathcal{D} \times \mathcal{D} \arrow[d, "\otimes"] \\
\mathcal{D} \times \mathcal{D} \arrow[r, "\otimes"'] & \mathcal{D}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathcal{D} \arrow[r, "\langle 1, e \rangle"] \arrow[dr, "1"'] & \mathcal{D} \times \mathcal{D} \arrow[d, "\otimes"] & \mathcal{D} \arrow[l, "\langle e, 1 \rangle"'] \arrow[dl, "1"] \\
& \mathcal{D} &
\end{tikzcd}
$$

where the double functor $1 : \mathcal{D} \to \mathcal{D}$ is the identity, the double functor $e : \mathcal{D} \to \mathcal{D}$ (with some abuse of the notation) is the constant double functor which associates the object e and identities on e respectively with each object and each morphism/cell of $\mathcal{D}$, and $\langle \_, \_ \rangle$ denotes the pairing of double functors induced by the cartesian product of double categories.

The commuting diagrams state that the tensor product $\_ \otimes \_$ is associative on both objects, arrows and cells, and that e is the unit for $\_ \otimes \_$.

Definition 5. (Category sMDCat) A *monoidal double functor* is a double functor which preserves tensor product and unit object. We denote by **sMDCat** the category of monoidal double categories and monoidal double functors.

Let $X : \mathcal{D} \times \mathcal{D} \to \mathcal{D} \times \mathcal{D}$ be the double functor which swaps the arguments, i.e., such that for each $A, B \in \mathcal{D}$, $X(A, B) = (B, A)$. In the one-dimensional case, a *symmetry* is a natural isomorphism between the tensor product $\__1 \otimes \__2$ (the functor $\otimes$) and the swapped tensor product $\__2 \otimes \__1$ (the functor $X; \otimes$) which satisfies some additional coherence axioms (MacLane 1971). A *double symmetry* is a generalized natural transformation, with a generalized inverse, and it satisfies some similar axioms.

Definition 6. (Symmetric strict Monoidal Double Categories) A *symmetric strict monoidal double category*, SsMD for short, is a tuple $(\mathcal{D}, \otimes, e, \sigma)$ such that the triple $(\mathcal{D}, \otimes, e)$ is a sMD, and σ is the generalized natural transformation pictured below.

$$\begin{tikzcd} \otimes \arrow[r, "\gamma"] \arrow[d, "\rho"'] \arrow[dr, phantom, "\sigma"] & X;\otimes \arrow[d, "1"] \\ X;\otimes \arrow[r, "1"'] & X;\otimes \end{tikzcd}$$

Moreover, the Kelly-MacLane *coherence axioms*‡ for γ, ρ and σ must hold:

1 For any vertical arrows $v : a \to c$, $u : b \to d$, and $w : a' \to c'$ in $\mathcal{D}$,

$$\gamma_{u \otimes w, v} = (1_u \otimes \gamma_{w,v}) * (\gamma_{u,v} \otimes 1_w), \qquad \gamma_{v,u} * \gamma_{u,v} = 1_{v \otimes u}.$$

2 For any horizontal arrows $h : a \to b$, $g : c \to d$, and $f : a' \to b'$ in $\mathcal{D}$,

$$\rho_{f \otimes g, h} = (1^f \otimes \rho_{g,h}) \cdot (\rho_{f,h} \otimes 1^g), \qquad \rho_{h,g} \cdot \rho_{g,h} = 1^{h \otimes g}.$$

3 For any objects a, b, and c in $\mathcal{D}$,

$$\sigma_{a \otimes b, c} = (1_a \otimes \sigma_{b,c}) \triangleleft (\sigma_{a,c} \otimes 1_b), \qquad \sigma_{a,b} \triangleleft \sigma_{b,a} = 1_{a \otimes b}.$$

The fact that σ is a generalized natural transformation means that all the following equations are satisfied:

— *Naturality* of γ and ρ:
For any pair of cells A, A' in $\mathcal{D}$,

‡ Translating the more general definition of symmetric monoidal category into the special case of symmetric *strict* monoidal category it could seem that also the axiom $\gamma_{id_e, v} = 1_v$ and the similar ones for ρ and σ should be stated. However, it is immediate to show that the other axioms are sufficient to satisfy this constraint. In fact, since $e \otimes e = e$, it follows that $\gamma_{id_e, v} = \gamma_{id_e \otimes id_e, v} = (1_e \otimes \gamma_{id_e, v}) * (\gamma_{id_e, v} \otimes 1_e) = \gamma_{id_e, v} * \gamma_{id_e, v}$. Thus, composing with γ_{v, id_e}, we obtain $1_{id_e \otimes v} = \gamma_{id_e, v}$. Finally, recalling that $id_e \otimes v = v$, we can conclude that $\gamma_{id_e, v} = 1_v$.

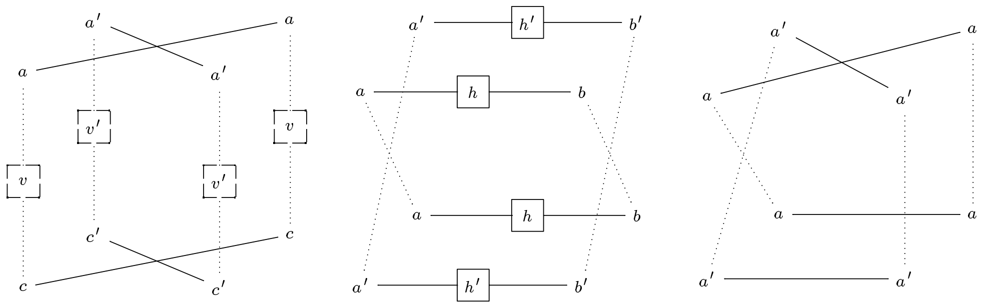


Fig. 22. The components $\gamma_{v,v'}$, $\rho_{h,h'}$ and $\sigma_{a,a'}$ of double symmetries: Graphically.

$$
\begin{tikzcd}
a\otimes a' \arrow[r, "\gamma_{a,a'}"] \arrow[d, "v\otimes v'"'] \arrow[dr, phantom, "\gamma_{v,v'}"] & a'\otimes a \arrow[d, "v'\otimes v"] \\
c\otimes c' \arrow[r, "\gamma_{c,c'}"'] & c'\otimes c
\end{tikzcd}
\qquad
\begin{tikzcd}
a\otimes a' \arrow[r, "h\otimes h'"] \arrow[d, "\rho_{a,a'}"'] \arrow[dr, phantom, "\rho_{h,h'}"] & b\otimes b' \arrow[d, "\rho_{b,b'}"] \\
a'\otimes a \arrow[r, "h'\otimes h"'] & b'\otimes b
\end{tikzcd}
\qquad
\begin{tikzcd}
a\otimes a' \arrow[r, "\gamma_{a,a'}"] \arrow[d, "\rho_{a,a'}"'] \arrow[dr, phantom, "\sigma_{a,a'}"] & a'\otimes a \arrow[d, "id_{a'}\otimes id_a"] \\
a'\otimes a \arrow[r, "id_{a'}\otimes id_a"'] & a'\otimes a
\end{tikzcd}
$$

Fig. 23. The components $\gamma_{v,v'}$, $\rho_{h,h'}$ and $\sigma_{a,a'}$ of double symmetries: Algebraically.

$$
\begin{tikzcd}
a \arrow[r, "h"] \arrow[d, "v"'] \arrow[dr, phantom, "A"] & b \arrow[d, "u"] \\
c \arrow[r, "g"'] & d
\end{tikzcd}
\qquad
\begin{tikzcd}
a' \arrow[r, "h'"] \arrow[d, "v'"'] \arrow[dr, phantom, "A'"] & b' \arrow[d, "u'"] \\
c' \arrow[r, "g'"'] & d'
\end{tikzcd}
$$

$$(A\otimes A') * \gamma_{u,u'} = \gamma_{v,v'} * (A'\otimes A) \qquad (A\otimes A')\cdot \rho_{g,g'} = \rho_{h,h'}\cdot (A'\otimes A).$$

— *Functoriality* of γ and ρ:
For any vertical arrows $v : a \to c$, $v' : a' \to c'$, $w : c \to d$, and $w' : c' \to d'$ in $\mathcal{D}$,

$$\gamma_{v\cdot w, v'\cdot w'} = \gamma_{v,v'} \cdot \gamma_{w,w'}.$$

For any horizontal arrows $h : a \to b$, $h' : a' \to b'$, $f : b \to c$, and $f' : b' \to c'$ in $\mathcal{D}$,

$$\rho_{h*f, h'*f'} = \rho_{h,h'} * \rho_{f,f'}.$$

For any pair of objects a and a' in $\mathcal{D}$:

$$\gamma_{id_a, id_{a'}} = 1^{\gamma_{a,a'}}, \qquad \rho_{id_a, id_{a'}} = 1_{\rho_{a,a'}}.$$

— *Naturality* of σ:
For any vertical arrows $v : a \to c$ and $u : b \to d$ in $\mathcal{D}$,

$$\gamma_{v,u}\cdot \sigma_{c,d} = \sigma_{a,b}\cdot 1_{u\otimes v}.$$

For any horizontal arrows $h : a \to b$ and $g : c \to d$ in $\mathcal{D}$,

$$\rho_{h,g} * \sigma_{b,d} = \sigma_{a,c} * 1^{g \otimes h}.$$

The generalized inverse of $\sigma_{a,b}$ can be easily defined in terms of σ, γ and ρ as follows:

— the $*$-*inverse* of $\sigma_{a,b}$ is $\sigma^*_{a,b} = \sigma_{b,a} \cdot 1_{\rho_{a,b}}$,

— the $\cdot$-*inverse* of $\sigma_{a,b}$ is $\sigma^{\cdot}_{a,b} = \sigma_{b,a} * 1^{\gamma_{a,b}}$, and

— the *generalized inverse* of $\sigma_{a,b}$ is $\sigma^{-1}_{a,b} = 1_{\rho_{b,a}} \cdot (\sigma_{a,b} * 1^{\gamma_{b,a}}) = 1^{\gamma_{b,a}} * (\sigma_{a,b} \cdot 1_{\rho_{b,a}})$.

Remark 3. The above notation could be somehow misleading, because the generalized inverse $\sigma^{-1}_{a,b}$ of $\sigma_{a,b}$ is not $\sigma_{b,a}$ as one might expect from the second coherence axiom for σ. The fact is that $\sigma_{a,b} * 1^{\gamma_{b,a}} \neq 1_{\rho_{a,b}}$; thus, $1^{\gamma_{b,a}} \neq \sigma^*_{a,b}$. In some sense, the cell $\sigma_{b,a}$ is just the diagonal inverse of $\sigma_{a,b}$.

Definition 7. (Category SsMDCat) We denote by **SsMDCat** the category of SsMD's and monoidal double functors preserving all the symmetries.

Proposition 1. The forgetful functor from **SsMDCat** to **Set**, mapping a SsMD into its set of objects has a left adjoint which maps each set S into the *free SsMD* on S (denoted by $DSym_S$) whose objects are the elements of the free monoid $S^{\otimes}$ over S.

The following *representation* theorem states the correspondence between double symmetries and ordinary symmetries.

Theorem 1. For any set S, the double category $DSym_S$ is isomorphic to the double category of quartets over the free symmetric strict monoidal category Sym_S on S.

Therefore, we have an interesting characterization of the components of σ as the possible square-shaped decompositions of the arrows of the free symmetric strict monoidal category Sym_S on S. An analogous result holds for the cartesian case (see Theorem 2). Note however that this second characterization is less informative than the one based on generalized transformations, because it is not evident how the auxiliary tiles can interact with all the other cells (tiles).

5. Cartesian Double Categories (with consistently chosen products)

A fairly general notion of double products should require the products to exist according to the four possible compositions that we have seen: horizontal ($\_ * \_$), vertical ($\_ \cdot \_$), and diagonal ($\_ \triangleleft \_$ and $\_ \triangleright \_$). We recall the usual definition of cartesian category.

Definition 8. (Terminal Object, Products, Cartesian Category) Let $\mathcal{C}$ be a category. We say that an object t of $\mathcal{C}$ is *terminal* if for any object c of $\mathcal{C}$ there is exactly one arrow from c to t.
We say that $\mathcal{C}$ has binary *products*, if for any pair of objects $a, b \in \mathcal{C}$ there exists an object u together with two *projections* $\Pi_a : u \to a$ and $\Pi_b : u \to b$ satisfying the following condition: for each object c and arrows $f : c \to a$ and $g : c \to b$ in $\mathcal{C}$ there exists a unique arrow $q : c \to u$ such that $f = q; \Pi_a$ and $g = q; \Pi_b$.
The category $\mathcal{C}$ has *canonical binary products* (also called *chosen binary products*) if a specific product diagram is given for each pair of objects.
The category $\mathcal{C}$ is *cartesian* if it has a terminal object and all (binary) products.

Pursuing the analogy with the one-dimensional case, we propose the following definitions

Definition 9. Given a double category $\mathcal{D}$, we say that $\mathcal{D}$ has all *double (binary) products* if the categories $\mathcal{D}^*$, $\mathcal{D}^{\cdot}$, $\mathcal{D}^{\triangleleft}$, and $\mathcal{D}^{\triangleright}$ all have (binary) products.
We say that $\mathcal{D}$ has a *double terminal object* if the categories $\mathcal{D}^*$, $\mathcal{D}^{\cdot}$, $\mathcal{D}^{\triangleleft}$, and $\mathcal{D}^{\triangleright}$ all have a terminal object.

Definition 10. (Cartesian Double Category) A double category $\mathcal{D}$ is called *cartesian* if it has all binary double products and double terminal objects.

However, we are interested in a much tighter notion of product, similar to the choice of a "canonical product". In fact, the more liberal definition does not establish any correspondence between the same notions on the different dimensions, but simply states their existence. Thus, we adopt the convention that not only are the products *chosen* in all the four dimensions, but that they are also *consistently chosen*. For simplicity, in the rest of the paper, we will consider only this kind of cartesian double categories if not stated otherwise, thus avoiding having to always specify that they have "consistently chosen products".

Remark 4. In general there are many different kinds of cartesian double categories with *partially chosen products*, where only some of the categories $\mathcal{D}^*$, $\mathcal{D}^{\cdot}$, $\mathcal{D}^{\triangleleft}$, and $\mathcal{D}^{\triangleright}$ have chosen products. In this sense, the more general definition could be called with *least chosen products* and the definition which we are about to discuss could be also called *with most chosen products*.

In the one-dimensional case, the notion of category with chosen products has an equivalent formulation in terms of symmetric monoidal category enriched by two natural transformations called *duplicator* and *discharger*. A *duplicator* is a natural transformation between the identity and the tensor product of two copies of the argument and satisfies some additional coherence axioms involving symmetries and dischargers. A *discharger* is a natural transformation between the identity and the constant functor mapping each element into the unit of the tensor product. Thus, *double duplicators* and *double dischargers* can be defined as generalized natural transformations satisfying similar coherence axioms. Let $\Delta : \mathcal{D} \to \mathcal{D} \times \mathcal{D}$ be the diagonal double functor which makes a copy of the argument, i.e., such that for each $A \in \mathcal{D}$, $\Delta(A) = (A, A)$. Then we have the following definition:

Definition 11. (Cartesian Double Categories with Consistently Chosen Products) A *cartesian double category (with consistently chosen products)* is a tuple

$$(\mathcal{D}, \otimes, e, \sigma, \pi, \tau, \phi, \psi)$$

such that $(\mathcal{D}, \otimes, e, \sigma)$ is a SsMD enriched with the generalized natural transformations π, τ, ϕ, and ψ pictured below.

$$\begin{tikzcd}
1_{\mathcal{D}} \arrow[r, "\nabla"] \arrow[d, "\delta"'] \arrow[dr, phantom, "\pi"] & \Delta;\otimes \arrow[d, "1"] \\
\Delta;\otimes \arrow[r, "1"'] & \Delta;\otimes
\end{tikzcd}
\quad
\begin{tikzcd}
1_{\mathcal{D}} \arrow[r, "1"] \arrow[d, "1"'] \arrow[dr, phantom, "\tau"] & 1_{\mathcal{D}} \arrow[d, "\delta"] \\
1_{\mathcal{D}} \arrow[r, "\nabla"'] & \Delta;\otimes
\end{tikzcd}
\quad
\begin{tikzcd}
1_{\mathcal{D}} \arrow[r, "!"] \arrow[d, "\dagger"'] \arrow[dr, phantom, "\phi"] & e \arrow[d, "1"] \\
e \arrow[r, "1"'] & e
\end{tikzcd}
\quad
\begin{tikzcd}
1_{\mathcal{D}} \arrow[r, "1"] \arrow[d, "1"'] \arrow[dr, phantom, "\psi"] & 1_{\mathcal{D}} \arrow[d, "\dagger"] \\
1_{\mathcal{D}} \arrow[r, "!"'] & e
\end{tikzcd}$$

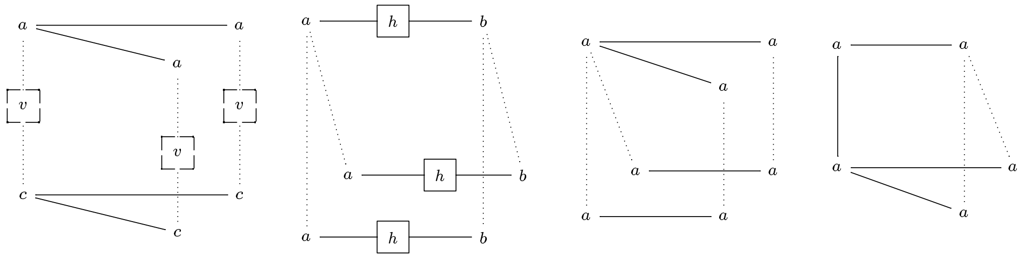


Fig. 24. The components ∇_v, δ_h, π_a and τ_a of double duplicators: Graphically.

Moreover the following equations must be satisfied:

1 Kelly-MacLane *coherence axioms* for ∇, $!$, δ, and $\dagger$:
For any vertical arrows $v : a \to c$, $u : b \to d$, and for any horizontal arrows $h : a \to b$, $g : c \to d$ in $\mathcal{D}$,

$$
\begin{array}{ll}
\nabla_{v\otimes u} = (\nabla_v \otimes \nabla_u) * (1_v \otimes \gamma_{v,u} \otimes 1_u), & \delta_{h\otimes g} = (\delta_h \otimes \delta_g) \cdot (1^h \otimes \rho_{h,g} \otimes 1^g), \\
!_{v\otimes u} = !_v \otimes !_u, & \dagger_{h\otimes g} = \dagger_h \otimes \dagger_g, \\
\nabla_{id_e} = 1_e = !_{id_e}, & \delta_{id_e} = 1_e = \dagger_{id_e}, \\
\nabla_v * (\nabla_v \otimes 1_v) = \nabla_v * (1_v \otimes \nabla_v), & \delta_h \cdot (\delta_h \otimes 1^h) = \delta_h \cdot (1^h \otimes \delta_h), \\
\nabla_v * \gamma_{v,v} = \nabla_v, & \delta_h \cdot \rho_{h,h} = \delta_h, \\
\nabla_v * (1_v \otimes !_v) = 1_v, & \delta_h \cdot (1^h \otimes \dagger_h) = 1^h.
\end{array}
$$

2 Kelly-MacLane *coherence axioms* for π, ϕ, τ, and ψ:
For any objects a and b in $\mathcal{D}$,

$$
\begin{array}{ll}
\pi_{a\otimes b} = (\pi_a \otimes \pi_b) \triangleleft (1_a \otimes \sigma_{a,b} \otimes 1_b), & \tau_{a\otimes b} = (\tau_a \otimes \tau_b) \triangleright (1_a \otimes \sigma^{-1}_{a,b} \otimes 1_b), \\
\phi_{a\otimes b} = \phi_a \otimes \phi_b, & \psi_{a\otimes b} = \psi_a \otimes \psi_b, \\
\pi_e = 1_e = \phi_e, & \tau_e = 1_e = \psi_e, \\
\pi_a \triangleleft (\pi_a \otimes 1_a) = \pi_a \triangleleft (1_a \otimes \pi_a), & \tau_a \triangleright (\tau_a \otimes 1_a) = \tau_a \triangleright (1_a \otimes \tau_a), \\
\pi_a \triangleleft \sigma_{a,a} = \pi_a, & \tau_a \triangleright \sigma^{-1}_{a,a} = \tau_a, \\
\pi_a \triangleleft (1_a \otimes \phi_a) = 1_a, & \tau_a \triangleright (1_a \otimes \psi_a) = 1_a.
\end{array}
$$

3 *double coherence axioms* for π, ϕ, τ, and ψ:
For any objects a in $\mathcal{D}$,

$$
\begin{array}{ll}
\tau_a * \pi_a = \nabla_a, & \tau_a \cdot \pi_a = \delta_a, \\
\psi_a * \phi_a = !_a, & \psi_a \cdot \phi_a = \dagger_a.
\end{array}
$$

The requirement for π, τ, ϕ and ψ to be generalized natural transformations implies that the following equations are satisfied:

— *Naturality* of ∇, $!$, $\dagger$, and δ:
For any cell A in $\mathcal{D}$,

$$
\begin{tikzcd}
a \arrow[r, "\nabla_a"] \arrow[d, "v"'] \arrow[dr, phantom, "\nabla_v"] & a \otimes a \arrow[d, "v \otimes v"] \\
c \arrow[r, "\nabla_c"'] & c \otimes c
\end{tikzcd}
\quad
\begin{tikzcd}
a \arrow[r, "h"] \arrow[d, "\delta_a"'] \arrow[dr, phantom, "\delta_h"] & b \arrow[d, "\delta_b"] \\
a \otimes a \arrow[r, "h \otimes h"'] & b \otimes b
\end{tikzcd}
\quad
\begin{tikzcd}
a \arrow[r, "\nabla_a"] \arrow[d, "\delta_a"'] \arrow[dr, phantom, "\pi_a"] & a \otimes a \arrow[d, "id_a \otimes id_a"] \\
a \otimes a \arrow[r, "id_a \otimes id_a"'] & a \otimes a
\end{tikzcd}
\quad
\begin{tikzcd}
a \arrow[r, "id_a"] \arrow[d, "id_a"'] \arrow[dr, phantom, "\tau_a"] & a \arrow[d, "\delta_a"] \\
a \arrow[r, "\nabla_a"'] & a \otimes a
\end{tikzcd}
$$

Fig. 25. The components ∇_v, δ_h, π_a and τ_a of double duplicators: Algebraically.

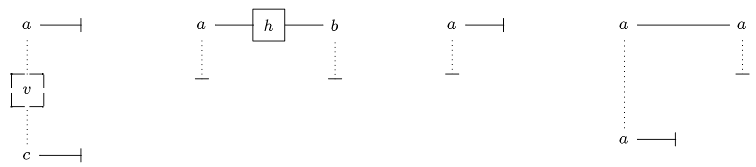


Fig. 26. The components $!_v$, $\dagger_h$, ϕ_a and ψ_a of double dischargers: Graphically.

$$
\begin{tikzcd}
a \arrow[r, "h"] \arrow[d, "v"'] \arrow[dr, phantom, "A"] & b \arrow[d, "u"] \\
c \arrow[r, "g"'] & d
\end{tikzcd}
$$

$$A * \nabla_u = \nabla_v * (A \otimes A), \qquad A *!_u =!_v, \qquad A \cdot \dagger_g = \dagger_h, \qquad A \cdot \delta_g = \delta_h \cdot (A \otimes A).$$

— *Functoriality* of ∇, $!$, $\dagger$, and δ:
For any vertical arrows $v : a \to c$ and $w : c \to d$ in $\mathcal{D}$,

$$\nabla_{v \cdot w} = \nabla_v \cdot \nabla_w, \qquad !_{v \cdot w} =!_v \cdot !_w.$$

For any horizontal arrows $h : a \to b$ and $f : b \to d$ in $\mathcal{D}$,

$$\delta_{h*f} = \delta_h * \delta_f, \qquad \dagger_{h*f} = \dagger_h * \dagger_f.$$

For any object a in $\mathcal{D}$,

$$\nabla_{id_a} = 1^{\nabla_a}, \qquad !_{id_a} = 1^{!_a}, \qquad \dagger_{id_a} = 1_{\dagger_a}, \qquad \delta_{id_a} = 1_{\delta_a}.$$

— *Naturality* of π, ϕ, ψ, and τ:
For any vertical arrow $v : a \to c$ in $\mathcal{D}$,

$$\nabla_v \cdot \pi_c = \pi_a \cdot 1_{v \otimes v}, \qquad !_v \cdot \phi_c = \phi_a, \qquad 1_v \cdot \psi_c = \psi_a, \qquad 1_v \cdot \tau_c = \tau_a \cdot \nabla_v.$$

For any horizontal arrow $h : a \to b$ in $\mathcal{D}$,

$$\delta_h * \pi_b = \pi_a * 1^{h \otimes h}, \qquad \dagger_h * \phi_b = \phi_a, \qquad 1^h * \psi_b = \psi_a, \qquad 1^h * \tau_b = \tau_a * \delta_h.$$

$$
\begin{tikzcd}
a \arrow[r, "!_a"] \arrow[d, "v"'] \arrow[dr, phantom, "!_v"] & e \arrow[d, "id_e"] \\
c \arrow[r, "!_c"'] & e
\end{tikzcd}
\quad
\begin{tikzcd}
a \arrow[r, "h"] \arrow[d, "\dagger_a"'] \arrow[dr, phantom, "\dagger_h"] & b \arrow[d, "\dagger_b"] \\
e \arrow[r, "id_e"'] & e
\end{tikzcd}
\quad
\begin{tikzcd}
a \arrow[r, "!_a"] \arrow[d, "\dagger_a"'] \arrow[dr, phantom, "\phi_a"] & e \arrow[d, "id_e"] \\
e \arrow[r, "id_e"'] & e
\end{tikzcd}
\quad
\begin{tikzcd}
a \arrow[r, "id_a"] \arrow[d, "id_a"'] \arrow[dr, phantom, "\psi_a"] & a \arrow[d, "\dagger_a"] \\
a \arrow[r, "!_a"'] & e
\end{tikzcd}
$$

Fig. 27. The components $!_v$, $\dagger_h$, ϕ_a and ψ_a of double dischargers: Algebraically.

Since most of the coherence axioms are either just a rephrasing of those for the one-dimensional case, we assume that only the last four *double coherence* axioms need some comment. They are needed in order to ensure the coherence of our structure (in the sense that *auxiliary* cells are uniquely identified by looking at their border). We should have also required similar axioms for σ and σ^{-1} in the definition of SsMD's, but since symmetries define isomorphisms there was no need to introduce explicitly σ^{-1}, because its existence was implied by the presence of σ, γ and ρ. Duplicators and dischargers (in general) are not isomorphisms; thus, we need to introduce both kinds of tiles: π and τ, and ϕ and ψ. In doing this we need to ensure that their compositions do not introduce any unnecessary additional structure. For instance, the vertical composition $\tau \cdot \pi$ returns a generalized natural transformation

$$
\begin{tikzcd}
1_{\mathcal{D}} \arrow[r, "1"] \arrow[d, "\delta"'] \arrow[dr, phantom, "\tau\cdot\pi"] & 1_{\mathcal{D}} \arrow[d, "\delta"] \\
\Delta;\otimes \arrow[r, "1"'] & \Delta;\otimes
\end{tikzcd}
$$

that is already present as the identity of δ.

Definition 12. (Category CDCat) We call **CDCat** the category of cartesian double categories with consistently chosen products and monoidal double functors preserving all the symmetries, duplicators and dischargers.

Proposition 2. The category **CDCat** is equivalent to the category of cartesian double categories (in the sense of Definition 10) where all the products are chosen and with morphisms product preserving functors on the nose.

Proposition 3. The forgetful functor from **CDCat** to **Set**, mapping each cartesian double category to its set of objects, has a left adjoint which maps each set S into the *free cartesian double category* on S (denoted by $DCart_S$) whose objects are the elements of the free monoid $S^{\otimes}$ over S.

Theorem 2. For any set S, the double category $DCart_S$ is isomorphic to the double category of quartets over the free cartesian category $Cart_S$ on S.

Likewise for the symmetric monoidal case, we have an alternative (intuitive) characterization for the auxiliary tiles of cartesian double categories. Still, the notion of generalized transformation seems to offer a more precise description of the interplay between auxiliary tiles and all the other cells in the double category.

6. Categorical semantics for Tile Logic

As we have explained in the introduction, the *tile model* (Gadducci and Montanari 2000) is a formalism for modular descriptions of concurrent systems, where the structural rules, called *tiles*, have a very general format. Tiles can be composed horizontally (through side effects), vertically (computational evolutions of a certain component), and in parallel (concurrent steps) to generate larger steps. By analogy with *rewriting logic* (Meseguer 1990; Meseguer 1992), where a logical theory is associated with a rewriting system in such a way that each computation is represented by a *sequent* entailed by the theory, the tile model also comes equipped with a purely logical presentation (Gadducci and Montanari 2000), where tiles become special sequents subject to certain inference rules. The sequents are also decorated with *proof terms*, and equivalence of proofs is then axiomatized in such a way that proofs decorating equivalent concurrent computations are identified. In this sense, *tile logic* is a logic of concurrent systems with synchronization mechanisms.

Process and *term tile logic* represent a wide class of systems whose configurations and observations can both be described as *net-process*-like structures and *terms* over a given signature. The fundamental concept which gives the basis of both logics, is the introduction of suitable *auxiliary* tiles, modeling all the consistent rearrangements of the interfaces. In (Bruni, Meseguer and Montanari 1998), as part of a more general study of the relationship between tile logic and rewriting logic (see concluding remarks), it has been proved that the axiomatization of proof sequents for process tile logic makes the algebra $A_p(\mathcal{R})$ of proof sequents into a SsMD, which is initial among the models of $\mathcal{R}$ in process tile logic. In this sense, the models of process tile logic could be adequately represented as symmetries-preserving monoidal double functors from $A_p(\mathcal{R})$ to generic SsMD's. Similarly, the axiomatization of proof sequents for term tile logic makes the algebra $A_t(\mathcal{R})$ of proof sequents into a cartesian double category, defining a suitable initial model of $\mathcal{R}$ for term tile logic.

For example, let us consider the process tile system $\mathcal{R}_{SMP} = \mathcal{R}_{SAMP} \cup \{\mathtt{synch}_a\}_{a\in\Gamma}$ defined in the introduction (the basic tiles are those presented in Figure 1 and 8). The algebra $A_p(\mathcal{R}_{SMP})$ consists in all the possible sequential and parallel compositions of the basic tiles of the system, the horizontal and vertical identities, and the double symmetries. Then, a possible solution to the synchronization problem proposed in the introduction for the configuration

$$\begin{array}{rcl} Q & = & (\mathit{proc}; \mathit{select}; (\mathit{id}_\circ \otimes (\mathit{select}; (\mathit{receive}_a \otimes \mathit{id}_\circ)))) \otimes \\ & \otimes & (\mathit{proc}; \mathit{select}; (\mathit{id}_\circ \otimes (\mathit{select}; (\mathit{send}_a \otimes \mathit{id}_\circ)))) \end{array}$$

can be expressed by the decorated sequent:

$$\begin{array}{rcl} S & = & (1^{R\otimes R} * 1^{\mathit{id}_\circ\otimes \mathit{receive}_a \otimes \mathit{id}_\circ \otimes \mathit{id}_\circ \otimes \mathit{send}_a \otimes \mathit{id}_\circ} * (1_\circ \otimes \sigma^{-1}_{\circ\otimes\circ,a} \otimes 1_{a\otimes\circ})) \cdot \\ & \cdot & (S' \otimes S' * (1_\circ \otimes \gamma_{\mathit{signal},\mathit{id}_\circ\otimes \mathit{id}_\circ} \otimes 1_{\mathit{signal}} \otimes 1_\circ) * (1_{\circ\otimes\circ\otimes\circ} \otimes \mathtt{synch}_a \otimes 1_\circ)) \cdot \\ & \cdot & (1^{R\otimes R} * (1_\circ \otimes \gamma_{\mathit{id}_a,\mathit{id}_\circ\otimes \mathit{id}_\circ} \otimes 1_{a\otimes\circ}) * 1^{\mathit{id}_\circ\otimes\circ\otimes\circ\otimes \mathit{receive}_a \otimes \mathit{send}_a \otimes \mathit{id}_\circ} * \\ & & *(1_\circ \otimes \sigma^{-1}_{a,\circ\otimes\circ} \otimes 1_{a\otimes\circ})) \end{array}$$

where $R = proc; select; (id_\circ \otimes select)$ and $S' = \mathtt{comm} * \mathtt{right} * (1_\circ \otimes \mathtt{left})$. Moreover, the axioms of $A_p(\mathcal{R}_{SMP})$ suffice to show the equivalence between S and the sequent T defined below:

$$\begin{aligned} T \ = \ & (1^{R\otimes R} * 1^{id_\circ \otimes receive_a \otimes id_\circ \otimes id_\circ \otimes send_a \otimes id_\circ} * (1_{\circ\otimes a} \otimes \sigma^{-1}_{a,\circ\otimes\circ} \otimes 1_\circ)) \cdot \\ \cdot \ & (S' \otimes S' * (1_\circ \otimes 1_{signal} \otimes \gamma_{id_\circ \otimes id_\circ, signal} \otimes 1_\circ) * (1_\circ \otimes \mathtt{synch}_a \otimes 1_{\circ\otimes\circ\otimes\circ})) \cdot \\ \cdot \ & (1^{R\otimes R} * (1_{\circ\otimes a} \otimes \gamma_{id_\circ \otimes id_\circ, id_a} \otimes 1_\circ) * 1^{id_\circ \otimes receive_a \otimes send_a \otimes id_{\circ\otimes\circ\otimes\circ}} * \\ & *(1_{\circ\otimes a} \otimes \sigma^{-1}_{\circ\otimes\circ,a} \otimes 1_\circ)) \end{aligned}$$

To see this, note that the only difference between S and T is given by the way in which the communicating components ($receive_a$ and $send_a$ are shifted closer (otherwise they are separated by two wires and the rule $\mathtt{sync}_a$ cannot be applied): In S we move the receive component near to the send component and then move it back after the synchronization; vice versa, in T, we swap the send component with the two wires.

Thus we have just to demonstrate the equality between S'' and T'' below, as the equivalence of S and T is then an easy consequence of functoriality and exchange law.

$$\begin{aligned} S'' \ = \ & (1^{receive_a \otimes id_\circ \otimes id_\circ \otimes send_a} * (\sigma^{-1}_{\circ\otimes\circ,a} \otimes 1_a)) \cdot \\ \cdot \ & ((\gamma_{signal, id_\circ \otimes id_\circ} \otimes 1_{signal}) * (1_{\circ\otimes\circ} \otimes \mathtt{synch}_a)) \cdot \\ \cdot \ & ((\gamma_{id_a, id_\circ \otimes id_\circ} \otimes 1_a) * 1^{id_{\circ\otimes\circ} \otimes receive_a \otimes send_a} * (\sigma^{-1}_{a,\circ\otimes\circ} \otimes 1_a)) \\ T'' \ = \ & (1^{receive_a \otimes id_\circ \otimes id_\circ \otimes send_a} * (1_a \otimes \sigma^{-1}_{a,\circ\otimes\circ})) \cdot \\ \cdot \ & ((1_{signal} \otimes \gamma_{id_\circ \otimes id_\circ, signal}) * (\mathtt{synch}_a \otimes 1_{\circ\otimes\circ})) \cdot \\ \cdot \ & ((1_a \otimes \gamma_{id_\circ \otimes id_\circ, id_a}) * 1^{receive_a \otimes send_a \otimes id_{\circ\otimes\circ}} * (1_a \otimes \sigma^{-1}_{\circ\otimes\circ,a})) \end{aligned}$$

First, let us concentrate on the "middle" subterm of S''

$$S''' = (\gamma_{signal, id_\circ \otimes id_\circ} \otimes 1_{signal}) * (1_{\circ\otimes\circ} \otimes \mathtt{synch}_a).$$

By naturality of symmetries we have:

$$1_{\circ\otimes\circ} \otimes \mathtt{synch}_a = \gamma_{id_\circ \otimes id_\circ, signal \otimes signal} * (\mathtt{synch}_a \otimes 1_{\circ\otimes\circ}) * \gamma_{synchro_a, id_\circ \otimes id_\circ}$$

Moreover, by coherence we know that:

$$\gamma_{id_\circ \otimes id_\circ, signal \otimes signal} = (\gamma_{id_\circ \otimes id_\circ, signal} \otimes 1_{signal}) * (1_{signal} \otimes \gamma_{id_\circ \otimes id_\circ, signal})$$

By coherence (and functoriality of tensor product) we also get:

$$(\gamma_{signal, id_\circ \otimes id_\circ} \otimes 1_{signal}) * (\gamma_{id_\circ \otimes id_\circ, signal} \otimes 1_{signal}) = 1_{signal \otimes id_\circ \otimes id_\circ \otimes signal}$$

Hence, $S''' = (1_{signal} \otimes \gamma_{id_\circ \otimes id_\circ, signal}) * (\mathtt{synch}_a \otimes 1_{\circ\otimes\circ}) * \gamma_{synchro_a, id_\circ \otimes id_\circ}$. To match the middle expression in T'' we just have to simplify $\gamma_{synchro_a, id_\circ \otimes id_\circ}$. This can be done by exploiting the (vertical) naturality of σ, obtaining after a few simplifications:

$$\gamma_{synchro_a, id_\circ \otimes id_\circ} = \sigma^*_{\circ\otimes\circ, a\otimes a} \cdot (1_{synchro_a} \otimes 1_{\circ\otimes\circ}) \cdot \sigma^{-1}_{\circ\otimes\circ, a\otimes a}$$

<table>
<tr><td colspan="2">$1_{a\otimes\circ\otimes\circ\otimes a}$</td><td>$\sigma^{-1}_{\circ\otimes\circ,a}\otimes 1_a$</td></tr>
<tr><td>$1_a\otimes\gamma_{\circ\otimes\circ,a}$</td><td>$1_a\otimes\gamma_{a,\circ\otimes\circ}$</td><td>$\sigma^{*}_{\circ\otimes\circ,a}\otimes 1_a$</td></tr>
<tr><td>$1_a\otimes\gamma_{\circ\otimes\circ,a}$</td><td>$1_a\otimes\sigma^{*}_{\circ\otimes\circ,a}$</td><td>$1_a\otimes\rho_{\circ\otimes\circ,a}$</td></tr>
</table>

Fig. 28.

Then, by using the exchange law and the properties of identities, we have:

$$\begin{array}{rcl} S''' & = & ((1_a \otimes \gamma_{id_\circ\otimes id_\circ,id_a}) * 1^{receive_a\otimes send_a\otimes id_\circ\otimes id_\circ} * \sigma^*_{\circ\otimes\circ,a\otimes a}) \cdot \\ & \cdot & ((1_{signal} \otimes \gamma_{id_\circ\otimes id_\circ,signal}) * (\mathtt{synch}_a \otimes 1_{\circ\otimes\circ}) * (1_{synchro_a} \otimes 1_{\circ\otimes\circ})) \cdot \\ & \cdot & ((1_a \otimes \gamma_{id_\circ\otimes id_\circ,id_a}) * 1^{receive_a\otimes send_a\otimes id_\circ\otimes id_\circ} * \sigma^{-1}_{\circ\otimes\circ,a\otimes a}) \end{array}$$

Now let us consider the upper subexpressions in S'' and T''. By naturality:

$$(1_a \otimes \gamma_{id_\circ\otimes id_\circ,id_a}) * 1^{receive_a\otimes send_a\otimes id_\circ\otimes id_\circ} = 1^{receive_a\otimes id_\circ\otimes id_\circ\otimes send_a} * (1_a \otimes \gamma_{id_\circ\otimes id_\circ,id_a})$$

Hence, if we show that

$$(\sigma^{-1}_{\circ\otimes\circ,a} \otimes 1_a) \cdot ((1_a \otimes \gamma_{id_\circ\otimes id_\circ,id_a}) * \sigma^*_{\circ\otimes\circ,a\otimes a}) = 1_a \otimes \sigma^{-1}_{a,\circ\otimes\circ}$$

then, we can horizontally compose both expressions with $1^{receive_a\otimes id_\circ\otimes id_\circ\otimes send_a}$ and then apply the exchange law to conclude that the upper subexpressions in S'' and T'' match.

By diagonal coherence of σ (and therefore of σ^*) we know that

$$\sigma^*_{\circ\otimes\circ,a\otimes a} = ((1_a \otimes \gamma_{id_a,id_\circ\otimes id_\circ}) * (\sigma^*_{\circ\otimes\circ,a} \otimes 1_a)) \cdot ((1_a \otimes \sigma^*_{\circ\otimes\circ,a}) * (1_a \otimes \rho_{id_\circ\otimes id_\circ,id_a}))$$

Then, the situation is illustrated in Figure 28. The diagram can be considerably simplified by noting that $\sigma^{-1}_{\circ\otimes\circ,a} \cdot \sigma^*_{\circ\otimes\circ,a} = 1_{a\otimes\circ\otimes\circ}$ (by definition of generalized inverse), $\gamma_{\circ\otimes\circ,a} * \gamma_{a,\circ\otimes\circ} = 1_{\circ\otimes\circ\otimes a}$ (by coherence), and $\sigma^*_{\circ\otimes\circ,a} * \rho_{\circ\otimes\circ,a} = \sigma^*_{\circ\otimes\circ,a}$. In fact the proof term can be rewritten to the expression $(1_a \otimes \gamma_{\circ\otimes\circ,a}) * (1_a \otimes \sigma^*_{\circ\otimes\circ,a}) = 1_a \otimes \sigma^{-1}_{a,\circ\otimes\circ}$, concluding this part of the proof.

Finally, we have to consider the lower subexpressions. The first thing to note is that $1^{id_\circ\otimes id_\circ\otimes receive_a\otimes send_a} = \gamma_{id_\circ\otimes id_\circ,id_a\otimes id_a} * 1^{receive_a\otimes send_a\otimes id_\circ\otimes id_\circ} * \gamma_{id_a\otimes id_a,id_\circ\otimes id_\circ}$ (by naturality), and that $\gamma_{id_\circ\otimes id_\circ,id_a\otimes id_a} = (\gamma_{id_\circ\otimes id_\circ,id_a} \otimes 1_a) * (1_a \otimes \gamma_{id_\circ\otimes id_\circ,id_a})$. Thus, by horizontally (pre)composing $1^{id_\circ\otimes id_\circ\otimes receive_a\otimes send_a}$ with $\gamma_{id_a,id_\circ\otimes id_\circ} \otimes 1_a$ we get:

$$(1_a \otimes \gamma_{id_\circ\otimes id_\circ,id_a}) * 1^{receive_a\otimes send_a\otimes id_\circ\otimes id_\circ} * \gamma_{id_a\otimes id_a,id_\circ\otimes id_\circ}$$

Thus, if we show that

$$\sigma^{-1}_{\circ\otimes\circ,a\otimes a} \cdot (\gamma_{id_a\otimes id_a,id_\circ\otimes id_\circ} * (\sigma^{-1}_{a,\circ\otimes\circ} \otimes 1_a)) = 1_a \otimes \sigma^{-1}_{\circ\otimes\circ,a}$$

then we can conclude the proof. We leave to the reader, with the help of Figure 29, the task of verifying that the equality above follows from the diagonal coherence of σ (and therefore of σ^{-1}).

$1_a \otimes \sigma^{-1}_{\circ\otimes\circ,a}$	$1_a \otimes \rho_{a,\circ\otimes\circ}$	$1_a \otimes \rho_{a,\circ\otimes\circ}$
$1_a \otimes \gamma_{a,\circ\otimes\circ}$	$\sigma^{-1}_{\circ\otimes\circ,a} \otimes 1_a$	$\gamma_{a,\circ\otimes\circ} \otimes 1_a$
$1_a \otimes \gamma_{a,\circ\otimes\circ}$	$\gamma_{a,\circ\otimes\circ} \otimes 1_a$	$\sigma^{-1}_{a,\circ\otimes\circ} \otimes 1_a$

Fig. 29.

Concluding Remarks

We have introduced the notion of *generalized natural transformation* between four double functors, showing that it extends the concept of natural transformation to the double category setting, in a more fairly way than the internal approach does. Our definition is equivalent to the notion of *hypertransformation* (Ehresmann 1963) when restricted to the 2-fold case. Then, we have defined the notion of *symmetric strict monoidal double category* and of *cartesian double category* (*with consistently chosen products*) pursuing the analogy with the one-dimensional case. As an important result, the generalized natural transformations that we have considered find an interesting characterization in terms of the square-shaped commuting diagrams involving their corresponding one-dimensional versions, expressed by Theorems 1 and 2. We also remark that the *Kelly-MacLane coherence axioms* can be lifted to our setting without much effort, thanks to the characterization of two suitable *diagonal categories*. An interesting theoretical application of symmetric and cartesian double categories is related to applications in the *Models of Computation* area, and in particular to a recently developed framework for the modular description of concurrent and distributed systems, called *tile logic* (Gadducci and Montanari 2000).

In the recent literature, tiles have been used with success for modeling several classes of applications, extending the SOS specification approach (Plotkin 1981) to *open* and *heterogeneous* systems. Applications range from *coordination languages*, where triggers and effects naturally represent coordination protocols, to *mobile processes*, where free and bound names and name extrusion must be handled. In (Montanari and Rossi 1999), a simple coordination model based on graph rewriting and synchronization is presented, using a class of tiles whose configurations are terms and whose observations are the elements of the free monoid over a set of basic observations. As another example, tile models for most process algebras (Gadducci and Montanari 2000) have process terms as configurations, and elements of the free monoid of actions (which are unary symbols) as observations.

The simplest possible interpretation of structured configurations and observations is considered in (Bruni and Montanari 2000), and consists in P/T net markings. As an important result, horizontal composition in the tile model yields a notion of *transition synchronization*, an important feature for compositionality, which is missing in ordinary nets (where only *token* synchronization is provided), and usually achieved through complex constructions.

In the future, it could be interesting to extend the framework presented here to weaker categories, where some of the naturality axioms for the generalized natural transformations are missing. Indeed, we are currently developing tile systems for a wide class of applications, where configurations and observations have only a *gs-monoidal* structure (Corradini and Gadducci 1997) rather than cartesian, i.e., where duplicators and dischargers are not natural. Indeed such structures are equipped with a notion of *name sharing* and have been shown useful to define a tile model for the (asynchronous) π-calculus (Milner, Parrow and Walker 1992), one of the most studied mobile calculi, in (Ferrari and Montanari 2000). In the same paper they have been used to represent both the operational and the abstract semantics of CCS with locations (Boudol, Castellani, Hennessy and Kiehn 1993) within the tile model. In both cases, the general notion of *tile bisimilarity* (Gadducci and Montanari 2000) is employed to quotient out configurations, thus recovering the ordinary abstract semantics. Moreover, although name extrusion and handling makes the transition system for the operational semantics of π-calculus infinitely branching and requires special notions of bisimulation, tiles equipped with term graphs handle names as wires, i.e., every sharing connection must be explicitly defined and explicit naming is not necessary. As a consequence, extrusion does not present any problem and can be specified using finite branching only. In the case of CCS with locations, sharing is used within configurations for modeling the operator $\_ \mid \_$ of process algebras, which in this context means sharing the same location. Within observations, sharing is used to express the fact that two events share the same cause, or, equivalently, that the same location has two different sublocations. A good notion of name sharing is also essential for other applications that we are currently developing, ranging from *logic programming* to λ-calculus. Other models should also be able to deal with co-duplicators and co-dischargers at the same time.

The categorical models developed here are also very important to pursue the comparison with similar models for rewriting logic, as we have done in (Bruni, Meseguer and Montanari 1998). In particular, we have defined an executable rewriting implementation of process and term tile systems whose correctness relies on a control mechanism over the rewritings. We have also shown how this mechanism can be specified using *internal strategies* in Maude (Clavel, Duran, Eker, Lincoln and Meseguer 1998) (a language based on rewriting logic that has been recently developed at SRI International).

Another possibility might be to extend our investigation to non-strict monoidal structures. Since in the one-dimensional case the non-strictness is expressed by three natural transformations, the intuition is that the lifting to the double case should just take into account suitable generalized natural transformations.

Acknowledgements We thank an anonymous referee for several helpful comments.

References

Boudol, G., Castellani, I., Hennessy, M. and Kiehn, A. (1993) Observing Localities. *TCS* **114**, 31–61.

Bruni, R. (1999) Ph.D. Thesis TD-99-01. Department of Computer Science, University of Pisa, Italy.

Bruni, R., Meseguer, J. and Montanari, U. (1998) Process and Term Tile Logic. Technical Report SRI-CSL-98-06, Computer Science Laboratory, SRI International, Menlo Park, California, USA. Also Technical Report TR-98-09, Department of Computer Science, University of Pisa, Italy.

Bruni, R. and Montanari, U. (2000) Zero-Safe Nets: Comparing the Collective and Individual Token Approaches. *Information and Computation* **156**, 46–89. Academic Press.

Clavel, M. G., Duran, F., Eker, S., Lincoln, P. and Meseguer, J. (1998) An Introduction to Maude (Beta Version). Computer Science Laboratory, SRI International, Menlo Park, CA, USA.

Corradini, A. and Gadducci, F. (1997) A 2-categorical Presentation of Term Graph Rewriting. In: E. Moggi, G. Rosolini (editors), *Proc. CTCS'97*, *LNCS* **1290**, 87–105.

Ehresmann, C. (1963) Catégories Structurées: I and II, Ann. Éc. Norm. Sup. **80**, 349–426, Paris; III, Topo. et Géo. diff. V, Paris.

M.C. Eekele, M.J. Plasmeijer, and M.R. Sleep, Eds., (1993) *Term Graph Rewriting: Theory and Practice*, Wiley, London.

Ferrari, G. L. and Montanari, U. (2000) Tile formats for located and mobile systems. *Information and Computation* **156**, 173–235. Academic Press.

Gadducci, F. (1996) On the Algebraic Approach to Concurrent Term Rewriting. Ph.D. Thesis TD-96-02. Department of Computer Science, University of Pisa, Italy.

Gadducci, F. and Montanari, U. (2000) The Tile Model. In: G. Plotkin, C. Stirling, M. Tofe, Eds., *Proof, Language and Interaction: Essays in Honour of Robin Milner*, MIT Press. Also Technical Report TR-96-27, Department of Computer Science, University of Pisa, Italy.

Grandis, M. and Paré, R. (1997) Limits in double categories. Dipartimento di Matematica, Universitá di Genova, Italy. Preprint **359**.

Kelly, G. M. and Street, R.H. (1974) Review of the Elements of 2-categories. *Lecture Notes in Mathematics* **420**, 75–103.

Larsen, K. G. and Xinxin, L. (1990) Compositionality Through an Operational Semantics of Contexts. In *Proc. ICALP'90*, *LNCS* **443**, 526–539.

MacLane, S. (1971) *Categories for the Working Mathematician*. Springer-Verlag.

Meseguer, J. (1990) Rewriting as a Unified Model of Concurrency. Tech. Report CSL-90-02R, Computer Science Laboratory, SRI International, Menlo Park, California, USA.

Meseguer, J. (1992) Conditional Rewriting Logic as a Unified Model of Concurrency. *TCS* **96**, 73–155.

Meseguer, J. and Montanari, U. (1998) Mapping Tile Logic into Rewriting Logic. In: F. Parisi-Presicce (editor), *Proc. 12th WADT Workshop on Algebraic Development Techniques, LNCS* **1376**, 62–91.

Milner, R., Parrow, J. and Walker, D. (1992) A Calculus of Mobile Processes (parts I and II). *Information and Computation* **100**, 1–77.

Montanari, U. and Rossi, F. (1999) Graph Rewriting, Constraint Solving and Tiles for Coordinating Distributed Systems. *Applied Categorical Structures* **7**(4), 333–370.

Plotkin, G. (1981) A Structural Approach to Operational Semantics. Technical Report DAIMI FN-19, Computer Science Department, Aarhus University.

Appendix A. Hypertransformations

As explained in Section 2.2, since double categories have two different notions of composition, it is not clear which of them should be used for a general notion of double

$$(\mathcal{A}) \quad \begin{tikzcd} a \arrow[d, "v"'] \\ b \arrow[d, "u"'] \\ c \end{tikzcd} \qquad\qquad \begin{tikzcd} \cdot \arrow[r, "\alpha(a)"] \arrow[d, "Pv"'] \arrow[dr, phantom, "\alpha(v)"] & \cdot \arrow[r, "\alpha'(a)"] \arrow[d, "Qv"] \arrow[dr, phantom, "\alpha'(v)"] & \cdot \arrow[d, "Rv"] \\ \cdot \arrow[r] \arrow[d, "Pu"'] \arrow[dr, phantom, "\alpha(u)"] & \cdot \arrow[r] \arrow[d, "Qu"] \arrow[dr, phantom, "\alpha'(u)"] & \cdot \arrow[d, "Ru"] \\ \cdot \arrow[r, "\alpha(c)"'] & \cdot \arrow[r, "\alpha'(c)"'] & \cdot \end{tikzcd} \quad (\mathcal{D})$$

Fig. 30. $\mathcal{D}$-wise transformations α and α'.

natural transformation. Ehresmann (1963) noticed that natural transformations can be expressed in terms of functors toward higher fold categories. The key point is that a natural transformation is in some sense *a functorial collection of commuting squares* (also called *quartets*). Therefore, to obtain a suitable notion of natural transformation for n-fold categories we have to consider quartets in all the n dimensions.

Definition 13. ($\mathcal{D}$-wise Transformation) Let $\mathcal{A}$ be a category, and let $\mathcal{D}$ be a double category. We denote by $T(\mathcal{D}, \mathcal{A})$ the category of *$\mathcal{D}$-wise transformations* from $\mathcal{A}$ to $\mathcal{D}$, whose objects are functors from $\mathcal{A}$ to the vertical 1-category $\mathcal{V}$ of $\mathcal{D}$ and whose composition law is deduced from that of the category $\mathcal{D}^*$, i.e., there is an arrow α from $P : \mathcal{A} \to \mathcal{V}$ to $Q : \mathcal{A} \to \mathcal{V}$ if and only if there is a functor $\alpha : \mathcal{A} \to \mathcal{D}^{\cdot}$ such that $\alpha; s^* = P$ and $\alpha; t^* = Q$, and given two arrows α and α' such that $\alpha; t^* = alpha'; s^*$ their composition is equal to the functor mapping each arrow v of $\mathcal{A}$ onto the cell $\alpha(v) * \alpha'(v)$ (see Figure 30).

The construction T extends to a functor as follows: given a double functor $F : \mathcal{D} \to \mathcal{E}$ and a functor $G : \mathcal{B} \to \mathcal{A}$ it suffices to define $T(F, G) = G; \alpha; F_1$, where the functor $F_1 : \mathcal{D}^{\cdot} \to \mathcal{E}^{\cdot}$ is the component of F relative to the vertical structure.

Example 2. Let $\mathbf{2}$ be the category with two objects, denoted by 0 and 1, and one arrow $z : 0 \to 1$ plus the identities. Then, for any double category $\mathcal{D}$, we have $T(\mathcal{D}, \mathbf{2}) \simeq \mathcal{D}^*$.

Proposition 4. Let $\mathcal{A}$ and $\mathcal{C}$ be two categories. The category $\mathcal{C}^{\mathcal{A}}$ whose objects are functors from $\mathcal{A}$ to $\mathcal{C}$ and whose arrows are natural transformations between such functors is isomorphic to the category $T(\square\ \mathcal{C}, \mathcal{A})$, where $\square\ \mathcal{C}$ is the double category of quartets of $\mathcal{C}$ (see Example 1). In fact, a functor from $\mathcal{A}$ to the vertical category of $\square\ \mathcal{C}$ (usually denoted by $\overset{\square}{\square}\mathcal{C}$) defines a natural transformation between functors from $\mathcal{A}$ to $\mathcal{C}$ and vice versa.

This notion can be generalized to n-fold categories by constructing the $2n$-fold category of quartets of quartets... (n times). Since double categories are 2-fold categories, this means that we need to define the 4-fold category of horizontal quartets of vertical quartets. Once a notion of *multiple functors* among categories of different folds has been given, then the notion of hypertransformation arises naturally as a multiple functor between the source n-fold category and the $2n$-fold category of quartets of quartets ... (n

times) generated by the target n-fold category. This means that a hypertransformation involving two n-fold categories $\vec{\mathcal{C}}$ and $\vec{\mathcal{D}}$ relates 2^n n-fold functors from $\vec{\mathcal{C}}$ to $\vec{\mathcal{D}}$.

Definition 14. (n-fold Category) Let n be a positive integer. An *n-fold category* $\vec{\mathcal{C}}$ is a tuple of n categories $(\mathcal{C}^1, \ldots, \mathcal{C}^n)$ with the same set of morphisms $\mathcal{C}$ satisfying the *permutability axiom*:

$(\mathcal{C}^i, \mathcal{C}^j)$ is a double category for each pair (i, j) of integers $1 \leq i < j \leq n$.

An element of $\mathcal{C}$ is called a *block*. The category $\mathcal{C}^i$ is called the *i-th category* of $\vec{\mathcal{C}}$, and its composition is denoted by $\_;_i\_$.

Notice that the set of objects of $\mathcal{C}^i$ defines a sub-category of $\mathcal{C}^j$ for each $i \neq j$.

Definition 15. (Category $\mathbf{Cat}_n$) If $\vec{\mathcal{C}}$ and $\vec{\mathcal{D}}$ are n-fold categories, a *n-fold functor* $F : \vec{\mathcal{C}} \to \vec{\mathcal{D}}$ is a map $F : \mathcal{C} \to \mathcal{D}$ that defines a functor $F : \mathcal{C}^i \to \mathcal{D}^i$ for each i. We denote by $\mathbf{Cat}_n$ the category whose objects are the small n-fold categories and whose morphisms are the n-fold functors between them.

By convention, a 0-fold category is just a set, and a 1-fold category is a category. Thus $\mathbf{Cat}_0$ is the category $\mathbf{Set}$, $\mathbf{Cat}_1$ is $\mathbf{Cat}$, and $\mathbf{Cat}_2$ is $\mathbf{DCat}$.

If γ is a permutation over the set $\{1, \ldots, n\}$ then $(\mathcal{C}^{\gamma(1)}, \ldots, \mathcal{C}^{\gamma(n)})$ is also an n-fold category and it is denoted by $\vec{\mathcal{C}}^{\gamma}$. If $i_1, \ldots, i_m$ is a sequence of m distinct elements of $\{1, \ldots, n\}$, we denote by $\mathcal{C}^{i_1, \ldots, i_m}$ the m-fold category $(\mathcal{C}^{i_1}, \ldots, \mathcal{C}^{i_m})$.

Definition 16. (Multiple Category) The category $\mathbf{MCat}$ of *multiple categories* is defined as follows. Its objects are the small n-fold categories, for every integer n. Let $\vec{\mathcal{C}}$ be an n-fold category and let $\vec{\mathcal{D}}$ be an m-fold category with $n \leq m$, then the morphisms $F : \vec{\mathcal{C}} \to \vec{\mathcal{D}}$ in $\mathbf{MCat}$, called *multiple functors*, are the n-fold functors F from $\vec{\mathcal{C}}$ to the n-fold category $(\mathcal{D}^1, \ldots, \mathcal{D}^n)$ (if $n > m$, then there is no morphism from $\vec{\mathcal{C}}$ to $\vec{\mathcal{D}}$).

Definition 17. (Internal Hom of MCat) Let $\vec{\mathcal{C}}$ be an n-fold category and $\vec{\mathcal{D}}$ an m-fold category. We denote by $Hom(\vec{\mathcal{C}}, \vec{\mathcal{D}})$ the *multiple category of multiple functors* from $\vec{\mathcal{C}}$ to $\vec{\mathcal{D}}$, which is defined as follows

— if $n > m$, then it is the empty set,
— otherwise (i.e., $n \leq m$), it is the $(m-n)$-fold category, on the set of multiple functors $F : \vec{\mathcal{C}} \to \vec{\mathcal{D}}$, whose i-th composition, for $1 \leq i \leq m - n$, is defined as $F(A);_{i+n} F'(A)$ if and only if the composite exists in $\mathcal{D}^{i+n}$ for each block A of $\mathcal{C}$.

Remark 5. For each pair (i, j) with $1 \leq i \leq m - n$ and $1 \leq j \leq n$, the category $Hom(\vec{\mathcal{C}}, \vec{\mathcal{D}})^i$ is a subcategory of the category of $(\mathcal{D}^j, \mathcal{D}^{i+n})$-wise transformations from $\mathcal{C}^j$ to $(\mathcal{D}^j, \mathcal{D}^{i+n})$. The permutability axiom is satisfied by $Hom(\vec{\mathcal{C}}, \vec{\mathcal{D}})$, since it is satisfied by $\vec{\mathcal{D}}$ and compositions are defined pointwise from that of $\vec{\mathcal{D}}$.

Example 3. If $\mathcal{A}$ and $\mathcal{C}$ are categories, then $Hom(\mathcal{A}, \square\, \mathcal{C})$ is the category $\mathcal{C}^{\mathcal{A}}$ of natural transformations between functors from $\mathcal{A}$ to $\mathcal{C}$.

Now, consider the category $\mathbf{2} \times \mathbf{2}$, and let $\mathcal{C}$ be a category. A functor $F : \mathbf{2} \times \mathbf{2} \to \mathcal{C}$ is entirely determined by the (commutative) square F of $\mathcal{C}$ (see Figure 31), because $F(z, z)$ is just the diagonal of this square. Moreover, every quartet of $\mathcal{C}$ can be obtained in this way. Thus, we can identify the set of functors $Hom(\mathbf{2} \times \mathbf{2}, \mathcal{C})$ with the set of quartets of $\mathcal{C}$ (which is also the set of blocks of $\square\, \mathcal{C}$).

$$
\begin{tikzcd}
(0,0) \arrow[r, "{(0,z)}"] \arrow[d, "{(z,0)}"'] \arrow[dr, "{(z,z)}"] & (0,1) \arrow[d, "{(z,1)}"] \\
(1,0) \arrow[r, "{(1,z)}"'] & (1,1)
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
\cdot \arrow[r, "{F(0,z)}"] \arrow[d, "{F(z,0)}"'] \arrow[dr, phantom, "F"] & \cdot \arrow[d, "{F(z,1)}"] \\
\cdot \arrow[r, "{F(1,z)}"'] & \cdot
\end{tikzcd}
$$

Fig. 31. Quartets as functors.

Let $\vec{\mathcal{C}}$ be a n-fold category, with $n > 1$, and consider the functors from $\mathbf{2} \times \mathbf{2}$ to its first category $\mathcal{C}^1$. It follows that, over the set of quartets of $\mathcal{C}^1$ we have not only the double category $\square\, \mathcal{C}^1$, but also the $(n-1)$-fold category $Hom(\mathbf{2} \times \mathbf{2}, \vec{\mathcal{C}})$ whose i-th composition is deduced pointwise from that of $\mathcal{C}^{i+1}$, i.e., given two quartets $Q = (A, B, C, D)$ and $Q' = (A', B', C', D')$ of $\mathcal{C}^1$ (this means that $A;_1 C = B;_1 D$ and $A';_1 C' = B';_1 D'$) then

$$Q;_i Q' = ((A;_{i+1} A'), (B;_{i+1} B'), (C;_{i+1} C'), (D;_{i+1} D'))$$

(if and only if the four compositions are defined in $\mathcal{C}^{i+1}$).

Definition 18. (Multiple Category of Quartets) The *multiple category of quartets* of $\vec{\mathcal{C}}$, denoted by $Sq\mathcal{C}$, is the $(n+1)$-fold category on the set of commuting squares of $\mathcal{C}^1$ such that: $(Sq\mathcal{C})^{1,\ldots,n-1} = Hom(\mathbf{2} \times \mathbf{2}, \vec{\mathcal{C}})$, $(Sq\mathcal{C})^n = \begin{smallmatrix}\square\\\square\end{smallmatrix}\mathcal{C}^1$ and $(Sq\mathcal{C})^{n+1} = \square\square\mathcal{C}^{n+1}$.

The previous construction induces a functor from $\mathbf{Cat}_n$ to $\mathbf{Cat}_{n+1}$ called *square* and denoted by $Sq_{n,n+1}$, mapping an n-fold functor $F : \vec{\mathcal{C}} \to \vec{\mathcal{D}}$ to the $(n+1)$-fold functor $SqF : Sq\mathcal{C} \to Sq\mathcal{D}$ such that

$$SqF(A, B, C, D) = (FA, FB, FC, FD).$$

A.1. *The 3-fold category $Sq\mathcal{D}$*

Given a double category $\mathcal{D} = (\mathcal{D}^{\cdot}, \mathcal{D}^*)$, the 3-fold category $Sq\mathcal{D}$ is defined as follows: its 2-nd and 3-rd categories are the vertical and horizontal categories $\begin{smallmatrix}\square\\\square\end{smallmatrix}\mathcal{D}^{\cdot}$ and $\square\square\mathcal{D}^{\cdot}$ of quartets of the first category $\mathcal{D}^{\cdot}$ of $\mathcal{D}$, and the first composition is deduced pointwise from that of $\mathcal{D}^*$ (see Figure 32).

A.2. *The 4-fold category $SqSq\mathcal{D}$*

The 4-fold category $SqSq\mathcal{D}$ is constructed as follows:

— the set of its blocks is $\square\ ((Sq\mathcal{D})^1)$, i.e., each block (Q_1, Q_2, Q_3, Q_4) is a quartet of $(Sq\mathcal{D})^1$, where $Q_i = (A_i, B_i, C_i, D_i)$ is a quartet of $\mathcal{D}^{\cdot}$ (i.e., A_i, B_i, C_i and D_i are cells of $\mathcal{D}$ and $A_i \cdot C_i = B_i \cdot D_i$) for $i = 1, \ldots, 4$. Thus, we can picture a block of $SqSq\mathcal{D}$ as the *frame* in Figure 33.

— The 1-st and 2-nd compositions are deduced pointwise from that of $\begin{smallmatrix}\square\\\square\end{smallmatrix}(\mathcal{D}^{\cdot})$ and $\square\square(\mathcal{D}^{\cdot})$, thus they consist in putting one frame *below* the other and one frame *inside* the other.

— The 3-rd and 4-th categories of $SqSq\mathcal{D}$ are the categories $\begin{smallmatrix}\square\\\square\end{smallmatrix}((Sq\mathcal{D})^1)$ and $\square\square((Sq\mathcal{D})^1)$, whose compositions are induced from that of $\mathcal{D}^*$ and consist in putting one frame

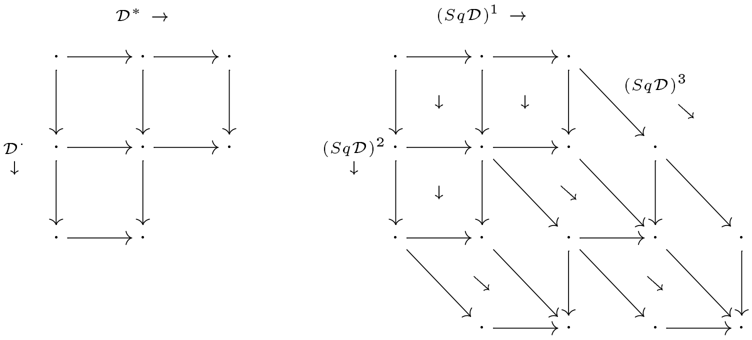


Fig. 32. The 3-fold category $Sq\mathcal{D}$ of a double category $\mathcal{D}$.

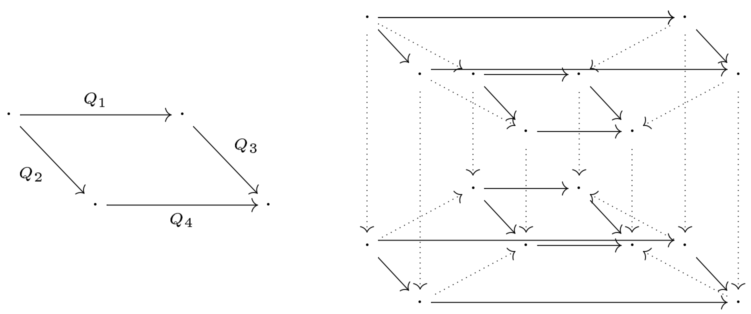


Fig. 33. A block in the 4-fold category $SqSq\mathcal{D}$ of a double category $\mathcal{D}$.

behind the other (i.e., the block in Figure 33 has source Q_1 and target Q_4) and one frame *beside* the other (source Q_2 and target Q_3).

A.3. *Hypertransformation*

We conclude by relating the definition of generalized natural transformation as given in Section 2.2 to the more general notion of *hypertransformation*.

Definition 19. (Hypertransformation) Let $\vec{\mathcal{C}}$ and $\vec{\mathcal{D}}$ be n-fold categories. Then consider the category $\square_n\vec{\mathcal{D}}$ which is obtained by applying n times the Sq construction and then swapping the order of the $2n$ categories in such a way that, for each $i = 1, ..., n$ the i-th category of $\square_n\vec{\mathcal{D}}$ is derived from $\overset{\square}{\square}(\mathcal{D}^i)$, and the $(i+n)$-th category of $\square_n\vec{\mathcal{D}}$ is derived from $\square\square(\mathcal{D}^i)$. Then $Hom(\vec{\mathcal{C}}, \square_n\vec{\mathcal{D}})$ is the category of *hypertransformations* from $\vec{\mathcal{C}}$ to $\vec{\mathcal{D}}$.

Let $\mathcal{D}$ and $\mathcal{E}$ be double categories. We denote by $\square_2\mathcal{E}$ the 4-fold category which is obtained by permuting the 2-nd and 3-rd compositions in $SqSq\mathcal{E}$. It follows that a multiple functor η from $\mathcal{D}$ to $\square_2\mathcal{E}$ maps objects of $\mathcal{D}$ into cells of $\mathcal{E}$, horizontal arrows of

$\mathcal{D}$ into horizontal commuting squares (of cells) of $\mathcal{E}$, vertical arrows of $\mathcal{D}$ into vertical commuting squares (of cells) of $\mathcal{E}$, and cells of $\mathcal{D}$ into commuting hypercubes of $\mathcal{E}$. The functoriality of η means that if two cells $A, B \in \mathcal{D}$ can be vertically (respectively horizontally) composed, then the hypercube ηB can be composed inside (respectively beside) ηA. Therefore, the collection of hypercubes $\{\eta A\}_{A \in \mathcal{D}}$ corresponds to the collection of commuting hypercubes of a generalized transformation, as illustrated in Section 2.2. In fact, the double category $Hom(\mathcal{D}, \Box_2\mathcal{E})$ can be described as follows:

— Its objects (i.e., vertices) are double functors from $\mathcal{D}$ to $\mathcal{E}$;
— Its horizontal arrows (i.e., objects for the 1-st category of $Hom(\mathcal{D}, \Box_2\mathcal{E})$) are natural $*$-transformations;
— Its vertical arrows (i.e., objects for the 2-nd category of $Hom(\mathcal{D}, \Box_2\mathcal{E})$) are natural $\cdot$-transformations;
— Its cells are generalized natural transformations between four double functors.

Finally, notice that compositions are deduced pointwise from the compositions of one frame *inside* the other and one frame *beside* the other (we remind that hypertransformations behave functorially w.r.t. the two compositions of one frame *below* the other and one frame *behind* the other).