Math. Struct. in Comp. Science: page 1 of 25.
doi:10.1017/S0960129517000019

A doctrinal approach to modal/temporal Heyting logic and non-determinism in processes

PAOLO BOTTONI[†], DANIELE GORLA[†], STEFANO KASANGIAN[‡] and ANNA LABELLA[†]

[†]*Dipartimento di Informatica, "Sapienza" Università di Roma, Rome, Italy*
Email: bottoni@di.uniroma1.it, gorla@di.uniroma1.it, labella@di.uniroma1.it
[‡]*Dipartimento di Matematica, Università di Milano, Milan, Italy*
Email: stefano.kasangian@gmail.com

Received 17 May 2016; revised 30 November 2016

The study of algebraic modelling of labelled non-deterministic concurrent processes leads us to consider a category L_B, obtained from a complete meet-semilattice **B** and from **B**-valued equivalence relations. We prove that, if **B** has enough properties, then L_B presents a two-fold internal logical structure, induced by two doctrines definable on it: one related to its families of subobjects and one to its families of regular subobjects. The first doctrine is Heyting and makes L_B a Heyting category, the second one is Boolean. We will see that the difference between these two logical structures, namely the different behaviour of the negation operator, can be interpreted in terms of a distinction between non-deterministic and deterministic behaviours of agents able to perform computations in the context of the same process. Moreover, the sorted first-order logic naturally associated with L_B can be extended to a modal/temporal logic, again using the doctrinal setting. Relations are also drawn to other computational models.

1. Introduction

For an agent A interacting with a system S, the latter appears to exhibit a non-deterministic behaviour if A can propose two identical sequences of actions eliciting different responses from S under identical circumstances. When reasoning on the possible behaviours of S, different logical inferences can then be drawn, depending on whether S is assumed to be deterministic or non-deterministic. Within the framework of our long-term research on the construction of a suitable categorical structure to model concepts from computer science, such as concurrency, bisimulation, non-determinism, temporal operators (Bottoni et al. 2012; De Nicola et al. 2010; Kasangian and Labella 1999), we set ourselves here to analyse the internal logics associated with the algebraic structures in which A's observations can be accounted for.

Starting from the temporal intuition underlying concurrent computational processes, a categorical structure was obtained, studied in Kasangian and Labella (1999) under the name *SymcatB*. Here, we study the algebraic and the internal logical structures of the category L_B, isomorphic to *SymcatB*, where **B** is the meet-semilattice of elementary observers carrying the temporal structure that provides a system of generators for L_B. We shift from *SymcatB* (where **B** plays the role of a categorical structure on which to

enrich in order to get $SymcatB$ (Walters 1981)) to L_B, with the aim of obtaining the same object in a language that is more immediate and more useful for comparisons. L_B is a generalization of the topos of presheaves (Ghezzi 2012) and it appears well suited to highlight mathematical phenomena connected with the notion of non-determinism. In particular, it captures, from the algebraic point of view, the specificity of Milner's version of observational non-determinism (Milner 1989).

The category L_B, although not a topos, has enough good properties and is particularly interesting with respect to the logic that can be canonically associated with it. Assuming that the logic of a mathematical structure depends on the doctrines one can define on it (Lawvere 1970), we find that L_B is naturally equipped with two doctrines, suitably related one to the other: a Heyting doctrine (called Sub), associated with the structure of its subobjects, and a Boolean doctrine (called $SSub$), associated with the subclass of regular subobjects. As expected, the particular structure of the Sub doctrine implies that the logic naturally associated with L_B is a Heyting one.

We will show that the difference between Sub and $SSub$ in L_B corresponds to the presence of non-determinism, in the sense discussed above, i.e., in terms of the observed behaviour of a process. Thus, the comparison between the two systems of doctrines gives us a measure of how much one has to weaken the logical system associated with L_B in order to take into account the non-determinism connected with the notion of observation in the subprocesses.

Following the doctrinal approach, we define other doctrines on L_B strictly related to the temporal doctrines introduced in Pisani (2010) (see Appendix B), according to the 'temporal nature' of **B**. Hence, the first-order logic naturally associated with L_B can be extended with modal/temporal operators in the style of Ghilardi and Meloni (1988) in a natural way. In the same way, a temporal doctrine is associated with the Boolean doctrine $SSub$ of strict subobjects.

Paper organisation. In Section 2, the category L_B is defined, and a sorted first-order logic is associated with it. L_B will be the Heyting category obtained from **B**, according to a specific criterion, and $SymcatB$, the category of **B**-enriched categories, will turn out to be isomorphic to it. In Section 3, the logic of L_B is extended with intrinsic modal/temporal operators introduced in terms of doctrines. Section 4 makes comparisons with some research lines related to the present one and Section 5 draws some conclusions.

2. The category of categories on a meet-semilattice B

In this section, we first define the algebraic structure of the category L_B and then the logics associated with it.

2.1. *The algebraic structure of* L_B

Let $\mathbf{B} = (B, \leqslant, \bigwedge)$ be a set B with a complete meet-semilattice structure, i.e., a poset with all possible small non-empty meets $\bigwedge$. Under these hypotheses, we have also binary meets (denoted by $\wedge$) and a minimum element $\bot = \bigwedge B$.

Definition 2.1 (Categories on a complete meet-semilattice B). Let $\mathbf{B} = (B, \leqslant, \bigwedge)$ be a complete meet-semilattice; we say that a category $\mathbf{C}$ is a *category on* $\mathbf{B}$ when

1. Each object $\mathcal{X}$ in $\mathbf{C}$ is a set X equipped with a $\mathbf{B}$-valued relation $\alpha_X : X \times X \to B$, such that $\forall x, y, z \in X$:
   — $\alpha_X(x, y) \wedge \alpha_X(y, z) \leqslant \alpha_X(x, z)$ (transitivity)
   — $\alpha_X(x, y) = \alpha_X(y, x)$ (symmetry).
2. A morphism in $\mathbf{C}$ is a function $f : \mathcal{X} \to \mathcal{Y}$ (induced by the function $f : X \to Y$ on the corresponding carrier sets) s.t. $\forall x, y \in X$:
   — $\alpha_X(x, x) = \alpha_Y(f(x), f(x))$.
   — $\alpha_X(x, y) \leqslant \alpha_Y(f(x), f(y))$.
3. Composition is function composition.

We say that a morphism f is *strict* if $\alpha_X(x, y) = \alpha_Y(f(x), f(y))$.

Intuitively, an object $\mathcal{X}$ is a set of B-labelled elements, allowed to be equal up to α_X. In fact, from Definition 2.1, we can derive the existence of a labelling function $\imath_X(x) = \alpha_X(x, x)$; $\imath_X$ can be seen as a special case of α_X, as it expresses the level of equality of an element with itself. In the following, we will explicitly mention $\imath$ to help intuition and simplify the mathematical treatment. With this in mind, one can immediately see that our definition is a special case of that of symmetric category on the bicategory associated with $\mathbf{B}$ in Walters (1981) (see Proposition 2.1). There, a classical example of meet-semilattice is considered, namely the one associated with the system of open sets in a topological space. We call $\imath_X$ the *extent* and α_X the *agreement* in $\mathcal{X}$ and we shall omit the subscript X when clear from the context or irrelevant. We call L_B the category on $\mathbf{B}$ having all the sets with a $\mathbf{B}$-valued relation as objects and all the possible functions preserving the $\mathbf{B}$-valued relation as morphisms.

Monomorphisms are in this case injective morphisms. A subobject, i.e., an equivalence class of monomorphisms, $\mathcal{X}'$ of $\mathcal{X}$ is a subset of $\mathbf{B}$-labelled elements in $\mathcal{X}$ such that $\alpha_{X'}$ is contained in α_X as $\mathbf{B}$-valued relation. It results from Definition 2.1 that L_B is concrete and has coproducts, while $\mathbf{B}$ is a system of generators for L_B in the sense of Borceux (1994).

We now point out some facts directly derivable Definition 2.1.

Proposition 2.1.

— $\alpha_X(x, y) \leqslant \imath_X(x) \wedge \imath_X(y)$
— The category $SymcatB$ of symmetric categories on $\mathbf{B}$ (Kasangian and Labella 1999; Walters 1981) is isomorphic to L_B. Indeed, $\mathbf{B}$ is given with a bicategory structure where objects are elements, 1-cells between two elements are elements contained in their intersection and 2-cells are given by the order relation.

Having assumed that $\mathbf{B}$ has all small non-empty meets $\bigwedge$, then it has also bounded joins, because we can define $\bigvee a_i = \bigwedge b_k$, where $a_i \leqslant b_k$ for every i and k. We require also that the following distributivity property is satisfied, i.e., binary meets do distribute over

joins (when they exist):

$$b \wedge \bigvee_{i \in I} a_i = \bigvee_{i \in I} (b \wedge a_i)$$

From now on, all generating semilattices will be distributive in the sense that finite meets distribute with respect to all the existing joins. Notice that general joins do not exist, not even in the finite case; so, our semilattice is not in general a lattice. Nonetheless, the existence of bounded joins is sufficient to prove Theorem 2.1.

We recall that a cartesian category **C** is *regular* if it has images stable under pullbacks; a regular category **C** is *coherent* if it has finite unions stable under pullbacks; a coherent, well-powered[†] category **C** is *geometric* if it has small unions stable under pullbacks and small intersections (Johnstone 2002). The main result of this section is Theorem 2.1, which makes use of the following Lemma.

Lemma 2.1. L_B is coherent.

Proof. We first need to define pullbacks, the terminal object, the image and the coproduct:

—*Pullback*: given $f : \mathcal{X} \to \mathcal{Z}$ and $g : \mathcal{Y} \to \mathcal{Z}$, we let $\mathcal{X} \times_Z \mathcal{Y} = \langle K, \iota, \alpha \rangle$, where $K = \{(x, y) \mid \iota_X(x) = \iota_Y(y), f(x) = g(y)\}$, $\iota(x, y) = \iota_X(x)$ and $\alpha((x, y), (x', y')) = \alpha_X(x, x') \wedge \alpha_Y(y, y')$.

—*Terminal object*: $\mathcal{B} = \langle B, id, \wedge \rangle$.

—*Image*: given $f : \mathcal{X} \to \mathcal{Y}$, we let $Im(f) = \langle K, \iota, \alpha \rangle$, where $K = \{y \in Y \mid \exists x \in X[y = f(x)]\}$, $\iota(y) = \iota_Y(y)$ and $\alpha(y, y') = \bigvee_{y'' \in Y} [\bigvee_{x \in f^{-1}(y), x'' \in f^{-1}(y'')} \alpha_X(x, x'') \wedge \bigvee_{x'' \in f^{-1}(y''), x' \in f^{-1}(y')} \alpha_X(x'', x')]$.

—*Coproduct*: $\mathcal{X} + \mathcal{Y} = \langle K, \iota, \alpha \rangle$, where $K = X \uplus Y$ ($\uplus$ represents disjoint union), $\iota(x) = \iota_X(x)$, $\iota(y) = \iota_Y(y)$, $\alpha(x, x') = \alpha_X(x, x')$, $\alpha(y, y') = \alpha_Y(y, y')$ and $\alpha(x, y) = \perp$, for $x, x' \in X$ and $y, y' \in Y$. This construction can be extended to arbitrary (set-indexed) coproducts.

Then, we should prove that what we have just defined is indeed the pullback, the terminal object, the image and the coproduct. This can be easily verified because we took the set-theoretical objects and imposed the correct equivalences in order to satisfy the universal properties. Notice that joins used to define images do always exist because they are bounded by $\alpha_Y(y, y')$. We are left to prove that images and unions (i.e., images of sums of monos) are stable under pullbacks. As usual, this is set-theoretically true. From the definition of equivalence and the fact that $\wedge$ preserves joins (the distributivity property), the same holds in L_B. The terminal object must be the sum of all generators with the maximal equivalence relation. □

Theorem 2.1. Let **B** be a complete meet-semilattice as above. Then, L_B is a Heyting category (Johnstone 2002) (originally called logos (Freyd and Scedrov 1990)); actually, L_B is a geometric category.

[†] A category is well powered whenever every object has a small poset of subobjects.

Proof. Given an object $\mathcal{X}$ in L_B, arbitrary $\cup$ and $\cap$ (the *join* and *meet*, resp.) can be defined between its subobjects, as usual, via pullback and image of coproducts, endowing them with the structure of a complete distributive lattice $(Sub(\mathcal{X}), \leqslant, \cup, \cap, 0)$. Given a morphism $f : \mathcal{X} \to \mathcal{Y}$, the image operator provides $\Sigma_f : Sub(\mathcal{X}) \to Sub(\mathcal{Y})$, which is left adjoint to the inverse image operator f^*. Unions are stable under meets (i.e., they are preserved by pullbacks). As the union exists for every family of elements in $Sub(\mathcal{X}), Sub(\mathcal{X})$ results into a complete Heyting algebra and f^* has a right adjoint Π_f as well (Freyd and Scedrov 1990, p.117). Then, L_B is a Heyting category and it is geometric. □

Strict monomorphisms in L_B coincide with regular monos in the sense that they are equalizers. In fact, every regular mono $\mathcal{X}' \to \mathcal{X}$ is a maximal subobject that equalizes two morphisms starting from its codomain; thus, it is necessarily strict. Vice versa, given a strict subobject of $\mathcal{X}$, we can construct $\mathcal{Y}$ as a set where all elements $x \in X$, except those belonging to X', are replicated. The replicas have the same extent as the original elements and are completely glued (maximal agreement) with them; hence, we can map $\mathcal{X}$ into $\mathcal{Y}$ in two different ways, obtaining a pair of morphisms with $\mathcal{X}' \to \mathcal{X}$ as equalizer. Strictness is preserved by identity, composition and pullbacks; images, though not necessarily strict, are strict if the original morphisms are, because agreement, already maximal, cannot change in the factorisation. In fact, in the case of strict morphisms, agreement in images is the same as in the codomain object.

We now consider, for every object $\mathcal{X}$, the structure $SSub(\mathcal{X})$ of its strict subobjects, i.e., subobjects associated with strict monos.

Proposition 2.2. For every $\mathcal{X}$ in L_B, the inclusion functor $i_X : SSub(\mathcal{X}) \to Sub(\mathcal{X})$ has a left-inverse left-adjoint functor $s_X : Sub(\mathcal{X}) \to SSub(\mathcal{X})$, which, in turn, has also a left adjoint j_X s.t. $j_X \dashv s_X \dashv i_X$.

Proof. s_X is the functor which makes a subobject $\mathcal{X}' = \langle X', \imath', \alpha' \rangle$ of $\mathcal{X} = \langle X, \imath, \alpha \rangle$ strict, i.e., $s_X \mathcal{X}' = \langle X', \imath', \alpha'' \rangle$, where $\alpha''(x', x'') = \alpha(x', x'')$. j_X is the functor which transforms a subobject $\mathcal{X}' = \langle X', \imath', \alpha' \rangle$ into $j_X \mathcal{X}' = \langle X', \imath', \bot \rangle$. By $\langle X', \imath', \bot \rangle$, we mean that in this subobject $\alpha(x, x') = \bot$ whenever $x \neq x'$. The proof of adjointness is routine. □

Proposition 2.3. For every $\mathcal{X}$, $SSub(\mathcal{X})$ has the structure of a Boolean algebra. Given a morphism $f : \mathcal{X} \to \mathcal{Y}$, there exist two operators $\Sigma^s_f, \Pi^s_f : SSub(\mathcal{X}) \to SSub(\mathcal{Y})$, left and right adjoint to the inverse image operator f^{*s}, respectively.

Proof. In this case α becomes irrelevant, so $\mathcal{X}$, with its $\imath$, is a B-labelled set and $SSub(\mathcal{X})$ is simply the powerset of $\mathcal{X}$ in Set/B, i.e., a Boolean algebra. We have that $f^{*s} s_Y = s_X f^*$. By multiplying on the right by i_Y and exploiting Proposition 2.2, we obtain $f^{*s} = s_X f^* i_Y$ and, therefore, we can define $\Sigma^s_f = s_X \Sigma_f j_X$ as the left adjoint to f^{*s}. On the other hand, $f^{*s} = s_X f^* j_Y$ also holds: in fact, applying s_X to $f^* j_Y$ and to $f^* i_Y$ makes them equal, because they are different only w.r.t. agreement and s_X makes us forget about it. Hence, we define $\Pi^s_f = s_Y \Pi_f i_X$ obtaining a right adjoint to $f^{*s} = s_X f^* j_Y$ (see Figure 1). □

Summing up, every object $\mathcal{X}$ in L_B is associated with two possible subobject structures, $Sub(\mathcal{X})$ and $SSub(\mathcal{X})$: the first one, obtained via all possible monos, enjoys a Heyting structure; the second one, obtained using only strict monos, enjoys a Boolean structure.

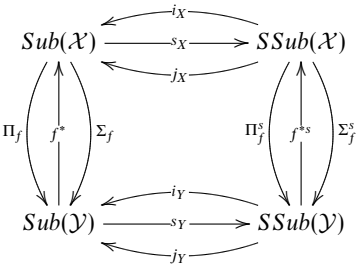

Fig. 1. The relationships between the two doctrines.

$SSub(\mathcal{X})$ is related to $Sub(\mathcal{X})$ via a pair of adjoint functors. The inclusion i_X does not preserve colimits, but its left adjoint s_X does, so $SSub(\mathcal{X})$ is a quotient algebra of $Sub(\mathcal{X})$. One could remark that s_X actually corresponds to double negation, as expectable, because the complement of a subobject always has the maximal possible agreement.

We now adapt a well-known definition that will play a crucial role. A *posetal hyperdoctrine* (Streicher 2003) is a functor $P : \mathbf{C}^{\mathbf{op}} \to \mathbf{pHa}$, where **pHa** is the category of pre-Heyting lattices, s.t. **C** has finite limits and, for every $f : J \to I$ in **C**, the functor $f^* = P(f) : P(I) \to P(J)$ has both left and right adjoint, $\exists_f$ and $\forall_f$, satisfying the Beck-Chevalley condition: for every pullback square $gq = fp$ in **C** and every $x \in P(J)$, the canonical morphisms $g^*\forall_f(x) \to \forall_q p^*(x)$ and $\exists_q p^*(x) \to g^*\exists_f(x)$ are isomorphisms. This condition is needed to guarantee that quantifiers preserve substitution.

Definition 2.2 (Doctrines). Let **C** be a regular category:

— A posetal hyperdoctrine $\mathcal{D}$ for **C** is a *Heyting* doctrine if it factorizes through **H**, the category of Heyting algebras with their homomorphisms.
— A Heyting doctrine $\mathcal{D}$ for **C** is *Boolean* if it factorizes through **Bool**, the category of Boolean algebras with their homomorphisms.
— A Heyting (resp. Boolean) *doctrine morphism* between two doctrines $\mathcal{D}$ and $\mathcal{D}'$ for **C** is a natural transformation from $\mathcal{D}$ to $\mathcal{D}'$ whose components commute with the operators that provide the Heyting (resp. Boolean) structure of the doctrines.

In the case of the functor *Sub*, the Beck–Chevalley condition is always satisfied because, being **C** regular, the existential functors commute with pullbacks up to isomorphisms and so do universal functors by the unicity of the adjoint (Johnstone 2002, Lemma A 1.4.11). However, if we consider a subfunctor $\mathcal{D}$ of *Sub* s.t. the pullback and the existential functors in $\mathcal{D}$ are simply the restriction of those in *Sub* (as it is the case for our *SSub*), the same result holds and can be proved via the same proof.

In this way, L_B comes naturally equipped with two doctrines: the first one, *Sub*, is Heyting and the second one, *SSub*, is Boolean. The quotient of the first one via *s* gives a Heyting morphism between them. A natural transformation $i : SSub \to Sub$ also exists, whose components are the functors i_X, but which fails to be a Heyting morphism.

Hence, the two doctrines are related as shown in Figure 1, where the diagram is commutative in the sense specified in Proposition 2.3.

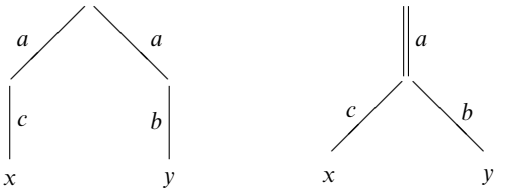

Fig. 2. Two trees with equal extents and different agreements.

2.2. *Labelled trees and beyond*

An example of our construction comes from the classical modelling of process semantics via Labelled Transition Systems (LTSs) (Aceto et al. 2007). In our framework, LTSs can be modelled by associating computations with paths (finite chains of moves) and by stating which prefixes of two different computations can be considered equal; the resulting structure has the shape of a tree.

Formally, given a set X and a meet-semilattice with all small non-empty meets $\mathbf{B} = (B, \leqslant, \bigwedge)$, with B to be thought of as a set of labels, we use the extent ι to label computations on $\mathbf{B}$. Hence, X will result into a set of computations of an LTS. In order to express up to which point two computations c_1 and c_2 are indistinguishable, we define their agreement to be a label $l \in B$ which is less than or equal to the meet of $\iota(c_1)$ and $\iota(c_2)$. From this definition, a notion of *generalised tree* derives, whose paths are labelled computations (Kasangian and Labella 1999) glued together as prescribed by the agreement. We call these objects generalised trees because, differently from usual trees, they also admit pathological situations, such as the empty tree, or trees where two paths are completely glued together. In this way, we have an instance of our construction L_B, often denoted $Tree_B$.

To support intuition, we consider trees labelled by some monoid on which the composition induces an order relation giving rise to a meet-semilattice structure with all small non-empty meets, typically a free monoid. In particular, let A^* be the free monoid on some alphabet A. An A^*-tree is usually called an A-labelled tree in the context of CCS (Milner 1989). A tree where all paths have different extent and pairwise have maximal agreement (given by $\wedge$) corresponds to the description of a *deterministic* behaviour of a process; otherwise, we are in presence of a non-deterministic behaviour (Milner 1989). Figure 2 shows an example of trees, each with two paths with equal extent (the words *ac* and *ab*) but different agreements: the agreement between the paths is the empty word in the leftmost tree and is the word *a* in the rightmost one.

In this case, morphisms are simulations between processes, in the sense that the behaviour of the codomain can simulate the behaviour of the domain. Indeed, there exists a morphism from the leftmost tree of Figure 2 to the rightmost one, but not vice versa: after performing an *a* in the leftmost one, an agent is no longer able to freely perform *b* or *c*, as would be possible in the rightmost one. An agent working under the assumption that the computations labelled via *ab* and *ac* are both allowed would then be 'surprised'

by the failure to perform the forbidden move on the path it is following. As usual (Milner 1989), the rightmost tree will be considered more deterministic than the leftmost one.

A strict monomorphism of codomain $\mathcal{X}$ represents a subobject (subprocess) with the same degree of non-determinism as $\mathcal{X}$, while a non-strict monomorphism of codomain $\mathcal{X}$ represents a subobject with more non-determinism than $\mathcal{X}$. This models the situation of an agent that is dealing with computations which are present in $\mathcal{X}$ as paths, but that cannot happen as in $\mathcal{X}$.

This example can be extended to elementary actions with different duration (Kasangian and Labella 1999). Given a (finite) alphabet A of atomic actions, a *partial piecewise constant function* (*pc-function*, for short) f is a partial function from $\mathbb{R}^+$ (the non-negative real numbers) to A, defined on a bounded interval $[0,t) \subseteq \mathbb{R}^+$ such that, for $a \in A$, $f^{-1}(a)$ is the union of finitely many intervals of the form $[r,s)$, with $0 \leqslant r < s$. If CA denotes the set of *pc*-functions, $(CA, \leqslant, \bigwedge)$ can be viewed as a complete meet-semilattice, where

i. $f \leqslant g$ *iff* $dom(f) \subseteq dom(g)$ and, if $x \in dom(f)$, then $f(x) = g(x)$.
ii. $\bigwedge_{i\in I} f_i$ w.r.t. the partial order above is defined as follows:
   a. $dom(\bigwedge_{i\in I} f_i) = [0,\bar{t})$, where $\bar{t}$ is s.t.
      — for all $x < \bar{t}$, $f_i(x)$ is defined for all $i \in I$ and $f_i(x) = f_j(x)$ for all $i,j \in I$;
      — if $f_i(\bar{t})$ is defined for all $i \in I$, then there are $i,j \in I$ s.t. $f_i(\bar{t}) \neq f_j(\bar{t})$.
   b. $(\bigwedge_{i\in I} f_i)(x) = f_i(x)$ for every $x \in dom(\bigwedge_{i\in I} f_i)$.

CA is the labelling structure for interleaving semantics in the *continuous time* case (Cardelli 1982). By replacing $\mathbb{R}^+$ by $\mathbb{N}$, one recovers the discrete case.

In all these examples, distributivity is guaranteed by the fact that bounded families are linearly ordered.

A different example is given by Mazurkiewicz traces (Kasangian and Labella 1999). A *monoid of traces* on a concurrent alphabet (A,I), where A is a set of events and I is a symmetric and irreflexive (independency) relation, is the monoid $(A^*,\cdot,\epsilon)/\equiv$, where $\equiv$ is the following congruence relation (Kasangian and Labella 1999):

$$s \equiv t \textit{ iff } \text{there is a sequence } \langle s_0, s_1, \ldots, s_n\rangle \text{ such that}$$

i. $s = s_0$,
ii. $t = s_n$ and
iii. for every $0 \leqslant i \leqslant n-1$, there are $u_i, v_i \in A^*$ such that $s_i = u_i abv_i, s_{i+1} = u_i bav_i$ and $(a,b) \in I$.

A trace $[s]$ is an equivalence class w.r.t. $\equiv$. $(A^*/\equiv, \leqslant, \bigwedge)$ is a semilattice where, given the traces $[u]$ and $[v]$, we put $[u] \leqslant [v]$ *iff* for every u_i in $[u]$ there is v_j in $[v]$ such that u_i is a prefix of v_j. The meet of a non-empty family of traces $\bigwedge[u]_k$ is the maximal common prefix of all the $[u]_k$. It is easy to see that, due to the discreteness of words, this is univocally determined. The associated structure is called **TA** and is the (Mazurkiewicz) trace labelling semilattice. Also, in this case distributivity can be proved because, in building the join of two traces (if it exists), we cannot prolong any of them with a suffix whose elements do not belong to a suffix of the other one.

One could also think of a temporal generalisation as in the previous case. Let us remark that **TA**-labelled trees, though trees in the abstract sense, are hardly depictable as such. They can model some forms of event structures.

2.3. *The logic of* L_B

A first-order logic can be easily associated with the category L_B. Let us first recall some well-known definitions from Johnstone (2002). Suppose we are given a suitable (w.r.t. L_B) first-order signature Σ (Johnstone 2002, Definition D 1.1.1) and we have recursively defined from it terms (Johnstone 2002, Definition D 1.1.2) and formulæ (Johnstone 2002, Definition D 1.1.3). The types of the language will be mapped into objects of the category; basic functional and predicate symbols are mapped to morphisms and subobjects, respectively, according to their type. We can interpret terms and formulæ of the language in L_B (which is a category with finite limits) in the usual way (Johnstone 2002; Pitts 2000), by fixing a *context* Γ, i.e., a finite set of typed variables containing all those that can appear free. Given a context, a term will be interpreted as a morphism from the product of the objects corresponding to the types of the variables to the object corresponding to the type of the term; a single variable is interpreted as the projection on its type. The interpretation of a formula ϕ, denoted by $[\![\phi]\!]$, is a subobject of the product of the objects corresponding to the types of the variables in the context (see Appendix A for a detailed definition). Given an interpretation and two formulae ϕ and ψ, a sequent $\phi \vdash \psi$ is satisfied *iff* $[\![\phi]\!]$ is a subobject of $[\![\psi]\!]$.

Definition 2.3 (Logics).

— A *Heyting* logic is a sorted language containing all the first-order logic formulae, equipped with a deductive system containing all the inference rules in Table 1 except for rule 8. An *infinitary Heyting* logic is a Heyting logic satisfying the infinitary versions of rules 3. and 4. in Table 1.

— A *Boolean (classical)* logic is a Heyting logic satisfying all the inference rules in Table 1. It is *infinitary* if it satisfies the infinitary versions of rules 3. and 4. in Table 1.

Proposition 2.4. L_B can interpret an infinitary first-order language with equality and its monos satisfy all the rules of an infinitary Heyting logic.

Proof. Since L_B is a geometric category with arbitrary intersections, it suffices to consider (Johnstone 2002), D 1.2. □

To conclude, we make the following considerations:

— If **B** is a trivial semilattice, L_B is equivalent to Set/B; in particular, L_B is equivalent to Set, if B is a singleton; all these categories are Boolean toposes.

— L_B contains a topos equivalent to $\hat{B}$ (Ghezzi 2012), the topos of presheaves over **B**. An object $\mathcal{X}$ of L_B is a presheaf if, together with an element x, it contains all its prefixes x' (i.e., all x' such that $\alpha(x, x') = \iota(x')$). In this way, $\mathcal{X}$ can be considered a presheaf on **B**, because restrictions can be easily defined. If **B** is the algebra of the open sets in a topological space, objects of L_B are the presentation of objects in the category of

Table 1. *Logical rules.*

1. identity $\frac{}{\phi \vdash_\Gamma \phi}$	substitution $\frac{\phi \vdash_\Gamma \psi}{\phi[s/y] \vdash_\Gamma \psi[s/y]}$ cut $\frac{\phi \vdash_\Gamma \psi \quad \psi \vdash_\Gamma \chi}{\phi \vdash_\Gamma \chi}$
2. equality $\frac{}{\top \vdash_\Gamma x{=}x}$	$\frac{}{x{=}y \wedge \phi \vdash_\Gamma \phi[y/x]}$
3. conjunction $\frac{}{\phi \vdash_\Gamma \top}$	$\frac{}{\phi \wedge \psi \vdash_\Gamma \phi}$ $\frac{}{\phi \wedge \psi \vdash_\Gamma \psi}$ $\frac{\phi \vdash_\Gamma \psi \quad \phi \vdash_\Gamma \chi}{\phi \vdash_\Gamma \psi \wedge \chi}$
4. disjunction $\frac{}{\bot \vdash_\Gamma \phi}$	$\frac{}{\phi \vdash_\Gamma \phi \vee \psi}$ $\frac{}{\psi \vdash_\Gamma \phi \vee \psi}$ $\frac{\phi \vdash_\Gamma \chi \quad \psi \vdash_\Gamma \chi}{\phi \vee \psi \vdash_\Gamma \chi}$
5. implication $\frac{\psi \vdash_\Gamma \phi \Rightarrow \chi}{\phi \wedge \psi \vdash_\Gamma \chi}$	$\frac{\phi \wedge \psi \vdash_\Gamma \chi}{\psi \vdash_\Gamma \phi \Rightarrow \chi}$
6. existential quantifier $\frac{\phi \vdash_{\Gamma, y:A} \psi}{\exists y{:}A \phi \vdash_\Gamma \psi}$	$\frac{\exists y{:}A \phi \vdash_\Gamma \psi}{\phi \vdash_{\Gamma, y:A} \psi}$
7. universal quantifier $\frac{\phi \vdash_{\Gamma, y:A} \psi}{\phi \vdash_\Gamma \forall y{:}A \psi}$	$\frac{\phi \vdash_\Gamma \forall y{:}A \psi}{\phi \vdash_{\Gamma, y:A} \psi}$
8. double negation $\frac{}{\neg\neg\phi \vdash_\Gamma \phi}$	

sheaves (Ghezzi 2012; Walters 1981). The inclusion functor from $\hat{B}$ to L_B has a left adjoint, namely the functor which 'closes' an object of L_B by adding the prefixes of all its elements.

As to the expressiveness of the logic associated with L_B, we can remark that if we assume that the prefix relation above is the interpretation of a predicate $\preccurlyeq$ in the language, then a first-order formula with both variables of type A such as

$$\forall x : A \; \exists y : A[x \preccurlyeq y]$$

expresses a liveness property in $\mathcal{X}$, where $\mathcal{X}$ is the interpretation of A.

Concerning provability, we know that

$$\forall x : A \; \forall y : A[\neg\neg(x \preccurlyeq y) \Rightarrow x \preccurlyeq y]$$

is true for strict monos, which enjoy a Boolean logic, but not in general for monos, which enjoy only a Heyting one. Of course negation and implication behave differently in the two contexts.

Another example of formula which is valid only for strict subobjects in the case of variables whose type is the terminal object $\mathcal{B}$ is

$$\forall x : \mathbf{1} \; \forall y : \mathbf{1}[\iota(x) \leqslant \iota(y) \Rightarrow x \preccurlyeq y]$$

where we identify ι with the unique morphism in L_B from a given subobject to $\mathcal{B}$. This is a crucial formula which marks the difference between deterministic and non-deterministic (in the sense of Milner (1989)) agents on L_B.

3. Modal/temporal logic

The introduction of modal operators is a typical technique to increase the expressive power of a language, e.g., in order to enable reasoning on bisimulation, fairness or liveness properties, etc. We will show now that if **B** is a non-trivial partially ordered set, then we can introduce into L_B a non-trivial intrinsic modal/temporal logic, still using the doctrinal approach. We define modal operators in L_B as endofunctors of $Sub\mathcal{X}$ (resp. $SSub\mathcal{X}$).

In doing this, we will make use of the notion of temporal doctrine in the sense of Pisani (2010) (see Appendix B) by simplifying it to cope with the special case when a general Heyting doctrine $\mathcal{D}$ is replaced by Sub or $SSub$ (i.e., attributes are subobjects, as explained later).

Definition 3.1 (Modal/temporal Doctrine). Given a Heyting category H and a Heyting doctrine $\mathcal{D}$ for H, a *modal/temporal doctrine for* $\mathcal{D}$ is a pair of Heyting doctrines $u\mathcal{D}$ and $d\mathcal{D}$ with two indexed functors i^u, i^d:

$$i^u : u\mathcal{D} \to \mathcal{D} \leftarrow d\mathcal{D} : i^d$$

satisfying the following properties for every object c in H:

— Functors i^u_c and i^d_c have both left and right adjoints

$$\Diamond^u_c \dashv i^u_c \dashv \Box^u_c$$

$$\Diamond^d_c \dashv i^d_c \dashv \Box^d_c$$

such that

$$\Diamond^u_c i^u_c \simeq id; \Box^u_c i^u_c \simeq id$$

$$\Diamond^d_c i^d_c \simeq id; \Box^d_c i^d_c \simeq id.$$

— Left adjoints satisfy the mixed Frobenius law in a natural way in c, i.e., for c' and c'' subobjects of c:

$$\Diamond^u_c(c' \times_c i^d_c(c'')) \simeq \Diamond^u_c(i^u_c \Diamond^u_c(c') \times_c i^d_c(c''))$$

$$\Diamond^d_c(c' \times_c i^u_c(c'')) \simeq \Diamond^d_c(i^d_c \Diamond^d_c(c') \times_c i^u_c(c'')).$$

A modal operator like $\Diamond$ (resp. $\Box$) is a closure (resp. interior) operator. In the example of trees discussed in Section 2.2, where the labelling monoid A^* could be thought of as bearing a time structure, these operators would assume the modal/temporal connotation well known in temporal logic. For a similar reason, with every object $\mathcal{X}$ in L_B, we associate the algebras $uSub\mathcal{X}$ and $dSub\mathcal{X}$ (resp. $uSSub\mathcal{X}$ and $dSSub\mathcal{X}$), alongside with $Sub\mathcal{X}$ (resp. $SSub\mathcal{X}$). These new algebras are obtained by closing the subobjects of $\mathcal{X}$ with respect to the prefix (resp. prolongation) relation (see Definition 3.2), i.e., by adding to the elements (paths) already existing in the subobject, the elements shorter (resp. longer) than them existing in $\mathcal{X}$. This amounts to adding the possibility of interrupting a path at a previous instant of time w.r.t. its end or to prolong it to a further instant of time, according to what is possible in $\mathcal{X}$.

Definition 3.2 (Prefix/prolongation/closedness).

— Given two elements x and x' in $\mathcal{X}$, we say that x is a *prefix* of x' (or x' is a *prolongation* of x), in symbols $x \leqslant x'$ (resp. $x' \geqslant x$), *iff* $\imath(x) = \alpha(x', x)$.
— $\mathcal{X}'$ is an up-closed (resp. down-closed) subobject of $\mathcal{X}$ *iff*, for every $x' \in \mathcal{X}'$ and $x \in \mathcal{X}$ s.t. $x \leqslant x'$ (resp. $x' \geqslant x$) in $\mathcal{X}$, then $x \in \mathcal{X}'$ and $x \leqslant x'$ (resp. $x' \geqslant x$) in $\mathcal{X}'$.
— $uSub\mathcal{X}$ ($dSub\mathcal{X}$) is the family of up-closed (resp. down-closed) subobjects of $\mathcal{X}$.
— $uSSub\mathcal{X}$ ($dSSub\mathcal{X}$) is the family of up-closed (resp. down-closed) strict subobjects of $\mathcal{X}$.

Fact 3.1. $\leqslant$ and $\geqslant$ are pre-order relations. When $\mathcal{X}$ is the terminal object, $\leqslant$ coincides with the order relation in the semilattice.

We first consider the easier case of strict subobjects, i.e., $SSub\mathcal{X}$. We can prove without difficulty that $uSSub\mathcal{X}$ and $dSSub\mathcal{X}$ are modal/temporal doctrines for $SSub\mathcal{X}$ (this will be proved in full detail later on for the more general – and difficult – non-strict case).

The modal/temporal operators are semantically defined as follows: given a subobject $\mathcal{X}'$ of $\mathcal{X}$ (for the sake of simplicity, when defining strict subobjects, we will identify extent and agreement in the subobject with those existing in the object, meaning that they actually are restrictions), we have

— $\spadesuit^s_{\mathcal{X}}\mathcal{X}' = \langle X' \cup \{x \in X \mid \exists x' \in X'[x \leqslant x']\}, \imath_{\spadesuit^s\mathcal{X}'} = \imath_{\mathcal{X}}, \alpha_{\spadesuit^s\mathcal{X}'} = \alpha_{\mathcal{X}}\rangle$
— $\triangle^s_{\mathcal{X}}\mathcal{X}' = \langle \{x' \in X' \mid \forall x \in X[x \leqslant x' \Rightarrow x \in X']\}, \imath_{\triangle^s\mathcal{X}'} = \imath_{\mathcal{X}}, \alpha_{\triangle^s\mathcal{X}'} = \alpha_{\mathcal{X}}\rangle$
— $\heartsuit^s_{\mathcal{X}}\mathcal{X}' = \langle X' \cup \{x \in X \mid \exists x' \in X'[x' \leqslant x]\}, \imath_{\heartsuit^s\mathcal{X}'} = \imath_{\mathcal{X}}, \alpha_{\heartsuit^s\mathcal{X}'} = \alpha_{\mathcal{X}}\rangle$
— $\triangledown^s_{\mathcal{X}}\mathcal{X}' = \langle \{x' \in X' \mid \forall x \in X[x' \leqslant x \Rightarrow x \in X']\}, \imath_{\triangledown^s\mathcal{X}'} = \imath_{\mathcal{X}}, \alpha_{\triangledown^s\mathcal{X}'} = \alpha_{\mathcal{X}}\rangle$.

$\spadesuit^s_{\mathcal{X}}\mathcal{X}'$ is the closure of $\mathcal{X}'$ with all the prefixes of its elements existing in $\mathcal{X}$; $\triangle^s_{\mathcal{X}}\mathcal{X}'$ is the maximum subobject of $\mathcal{X}'$ which results closed w.r.t. prefixes of its elements existing in $\mathcal{X}$; $\heartsuit^s_{\mathcal{X}}\mathcal{X}'$ is the closure of $\mathcal{X}'$ with all the prolongations of its elements existing in $\mathcal{X}$; $\triangledown^s_{\mathcal{X}}\mathcal{X}'$ is the maximum subobject of $\mathcal{X}'$ which results closed w.r.t. prolongations of its elements existing in $\mathcal{X}$.

Fact 3.2. $uSub\mathcal{X}$ and $dSub\mathcal{X}$ are posets w.r.t. the order of $Sub\mathcal{X}$; injections $i_X : uSub\mathcal{X} \to Sub\mathcal{X}$ are monotonic functions (functors).

In order to prove that from this situation a modal/temporal doctrine arises for $Sub\mathcal{X}$, one has to define suitable operators that can provide adjoints to injections, as required in Definition 3.1.

Definition 3.3 (Modal/temporal operators in L_B). We define in L_B the following operators:

— $\spadesuit_{\mathcal{X}}\mathcal{X}' = \langle X'', \imath_{\spadesuit\mathcal{X}'}, \alpha_{\spadesuit\mathcal{X}'}\rangle$ where
  – $X'' = X' \cup \{x \in X \mid \exists x' \in X'[x \leqslant x']\}$
  – $\imath_{\spadesuit\mathcal{X}'} = \imath_{\mathcal{X}}$,
  – $\alpha_{\spadesuit\mathcal{X}'}(x', x'') = \begin{cases} \alpha_{\mathcal{X}'}(x', x'') \vee \bigvee_{x \mid x \leqslant x', x \leqslant x''} \imath(x) & \text{if } x', x'' \in X' \\ \bigvee_{x \mid x \leqslant x', x \leqslant x''} \imath(x) & \text{otherwise} \end{cases}$
— $\triangle_{\mathcal{X}}\mathcal{X}' = \langle \{x' \in X' \mid \forall x \in X[x \leqslant x' \Rightarrow (x \in X' \wedge \alpha_{\mathcal{X}'}(x, x') = \imath_{\mathcal{X}}(x))]\}, \imath_{\mathcal{X}'}, \alpha_{\mathcal{X}'}\rangle$
— $\heartsuit_{\mathcal{X}}\mathcal{X}' = \langle X'', \imath_{\heartsuit\mathcal{X}'}, \alpha_{\heartsuit\mathcal{X}'}\rangle$ where

– $X'' = X' \cup \{x \in X \mid \exists x' \in X'[x' \leqslant x]\}$

– $\iota_{\heartsuit\mathcal{X}'} = \iota_{\mathcal{X}}$

– $\alpha_{\heartsuit\mathcal{X}'}(x', x'') = \begin{cases} \alpha_{\mathcal{X}}(x', x'') & \text{if } x', x'' \in X' \text{ and } \exists x \in X[x' \leqslant x, x'' \leqslant x] \\ \bigvee_{y', y'' \in X' : y' \leqslant x', y'' \leqslant x''} \alpha_{\mathcal{X}'}(y', y'') & \text{otherwise} \end{cases}$

(The second part includes also the case in which $x', x'' \in X'$ and they do not have a common prolongation).

— $\nabla_{\mathcal{X}}\mathcal{X}' = \langle \{x' \in X' \mid \forall x \in X[x' \leqslant x \Rightarrow (x \in X' \wedge \alpha_{\mathcal{X}'}(x, x') = \iota_{\mathcal{X}}(x))]\}, \iota_{\mathcal{X}'}, \alpha_{\mathcal{X}'} \rangle$.

In this general case, the operators have a meaning similar to the strict case, but for $\spadesuit_{\mathcal{X}}$ and $\heartsuit_{\mathcal{X}}$ one has to be careful in the definition of the agreement, as a naive definition would not lead to subobjects, or it would fail to define a functor. In the definition of agreement, one has to add the agreement due to the presence of new prefixes. The join expressing this does exist because it is bounded.

Lemma 3.1. All the operators in Definition 3.3 are monotonic functions (functors):

$$i^u_{\mathcal{X}} : uSub\mathcal{X} \to Sub\mathcal{X} \leftarrow dSub\mathcal{X} : i^d_{\mathcal{X}}$$

Moreover, $i^u_{\mathcal{X}}$ and $i^d_{\mathcal{X}}$ have both left and right adjoint, namely (forgetting the index $\mathcal{X}$): $\spadesuit \dashv i^u \dashv \triangle$ and $\heartsuit \dashv i^d \dashv \nabla$ such that

$$\spadesuit i^u \simeq id; \triangle i^u \simeq id$$

$$\heartsuit i^d \simeq id; \nabla i^d \simeq id.$$

Proof (Sketch). We have to prove that the definitions of $\spadesuit\mathcal{X}'$, $\triangle\mathcal{X}'$, $\heartsuit\mathcal{X}'$ and $\nabla\mathcal{X}'$ given above can be extended to functors, and then prove the adjunctions. All the operators are monotonic and adjunctions express the fact that they represent the minimal closure and the maximal closed subobject respectively. These properties and the required isomorphisms are obtained by verifying simple inequations between subobjects. □

Lemma 3.2. *uSub* and *dSub* can be extended to Heyting doctrines in L_B.

Proof. We first prove that *uSub* and *dSub* are Heyting algebras: due to the adjunctions from Lemma 3.1, *uSub* and *dSub* inherit unions and intersections from *Sub*. In fact $\spadesuit\mathcal{X}'$, being a left adjoint, preserves unions and, being a left inverse, makes them the same as in $Sub\mathcal{X}$. Dually, $\triangle\mathcal{X}'$, being a right adjoint, preserves intersections and, being a left inverse, makes them the same as in $Sub\mathcal{X}$. The same happens with *dSub*, using $\heartsuit\mathcal{X}'$ and $\nabla\mathcal{X}'$. Distributivity holds because it holds in $Sub\mathcal{X}$. Note that in both *uSub* and *dSub* the pseudo-complement is not the same as in $Sub(\mathcal{X})$, because closedness must be preserved in complementation, so that they are not Heyting subalgebras of $Sub\mathcal{X}$. We define: $\neg_u\mathcal{X}'$ as $\triangle(\neg\mathcal{X}')$.

Now it is sufficient to show that, given a morphism $f : \mathcal{X} \to \mathcal{Y}$, the inverse image operator f^* can be restricted to the appropriate family and it has left and right adjoints. Actually, due to the definition of morphism, a non-prefix can be mapped into a prefix, but not vice versa, while prefixes are preserved by f. Hence, f^* preserves up- (down-) closedness. f^* restricted to $uSub\mathcal{X}$ ($dSub\mathcal{X}$) has both adjoints, but they are not the

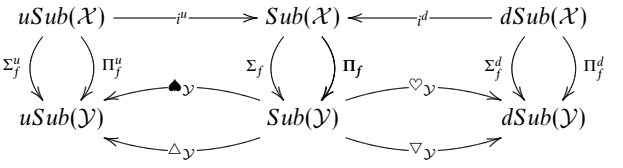

Fig. 3. Definition of left and right adjoints.

immediate restrictions of those existing in $Sub\mathcal{X}$: we need to compose the general ones with the suitable closure operators and restrict the result. Namely (see Figure 3),

$$\begin{aligned}\Sigma^u_f &= \spadesuit \circ \Sigma_f \circ i^u\\ \Pi^u_f &= \triangle \circ \Pi_f \circ i^u\\ \Sigma^d_f &= \heartsuit \circ \Sigma_f \circ i^d\\ \Pi^d_f &= \triangledown \circ \Pi_f \circ i^d.\end{aligned}$$

With these definitions, the adjointness result is routine. □

Theorem 3.1.

$$i^u : uSub \to Sub \leftarrow dSub : i^d$$

is a modal/temporal doctrine for *Sub*.

Proof. The proof proceeds from Lemmas 3.1 and 3.2. We are left to prove the mixed Frobenius laws. To this end, we notice that the left-hand subobject in their expression (see Definition 3.1) is contained in the right-hand one, since in the latter we have operated a double closure which could, possibly, increase elements and/or their agreement. Vice versa, if some prefix (prolongation) is lost in making the intersection on the right-hand side, it will be restored in making the external closure, so that also the other inclusion holds. □

Correspondingly, we can extend a first-order logic with four modal/temporal operators to be interpreted in the L_B operators above. Table 2 shows rules which hold in L_B. In other words, L_B provides us with two sorted Kripke-like semantics, as every type $\mathcal{X}$ can be thought of as a set of possible worlds, with $\preccurlyeq$ or $\succcurlyeq$ as the accessibility relation. Let p be a path (world) in $\mathcal{X}$, where we identify p with its terminal state. We define

— $p \models_{\mathcal{X}} \phi$ *iff* $p \in \mid \phi \mid$ (i.e., its terminal state enjoys ϕ), where $\mid \phi \mid$ is the interpretation of the formula ϕ.

— *Future accessibility* between paths is defined as the prefix relation above.

  – $p \models_{\mathcal{X}} \Diamond^u \phi$ *iff* $\exists q(p \preccurlyeq q \wedge q \models_{\mathcal{X}} \phi)$. This is equivalent to saying that $p \in \spadesuit_{\mathcal{X}} \mid \phi \mid$. In other words: 'there is a future of p when ϕ becomes true'.

  – $p \models_{\mathcal{X}} \Box^d \phi$ *iff* $\forall q(p \preccurlyeq q \Rightarrow q \models_{\mathcal{X}} \phi)$. This is equivalent to saying that $p \in \triangledown_{\mathcal{X}} \mid \phi \mid$. In other words: 'in all the possible futures of p, ϕ is true'.

— Dually, we define the *past accessibility* relation as the prolongation relation, so that

  – $p \models_{\mathcal{X}} \Diamond^d \phi$ *iff* $\exists q(q \preccurlyeq p \wedge q \models_{\mathcal{X}} \phi)$.

  – $p \models_{\mathcal{X}} \Box^u \phi$ *iff* $\forall q(q \preccurlyeq p \Rightarrow q \models_{\mathcal{X}} \phi)$.

Table 2. *Modal/temporal rules.*

μ_1) past possibility $\frac{}{\phi \vdash_\Gamma \Diamond^d \phi}$ future possibility $\frac{}{\phi \vdash_\Gamma \Diamond^u \phi}$

μ_2) past necessity $\frac{}{\Box^u \phi \vdash_\Gamma \phi}$ future necessity $\frac{}{\Box^d \phi \vdash_\Gamma \phi}$

μ_3) $\Diamond^d$ monotonicity $\frac{\phi \vdash_\Gamma \psi}{\Diamond^d \phi \vdash_\Gamma \Diamond^d \psi}$ $\Diamond^u$ monotonicity $\frac{\phi \vdash_\Gamma \psi}{\Diamond^u \phi \vdash_\Gamma \Diamond^u \psi}$

μ_4) $\Box^u$ monotonicity $\frac{\phi \vdash_\Gamma \psi}{\Box^u \phi \vdash_\Gamma \Box^u \psi}$ $\Box^d$ monotonicity $\frac{\phi \vdash_\Gamma \psi}{\Box^d \phi \vdash_\Gamma \Box^d \psi}$

μ_5) $\Box^u$ S4 $\frac{}{\Box^u \phi \vdash_\Gamma \Box^u \Box^u \phi}$ $\Box^d$ S4 $\frac{}{\Box^d \phi \vdash_\Gamma \Box^d \Box^d \phi}$

μ_6) $\Diamond^d$ S4 $\frac{}{\Diamond^d \Diamond^d \phi \vdash_\Gamma \Diamond^d \phi}$ $\Diamond^u$ S4 $\frac{}{\Diamond^u \Diamond^u \phi \vdash_\Gamma \Diamond^u \phi}$

μ_9) $\Diamond \bigvee$ preservation $\frac{}{\Diamond^u \bigvee_i \phi_i \vdash_\Gamma \bigvee_i (\Diamond^u \phi_i)}$ $\frac{}{\bigvee_i (\Diamond^u \phi_i) \vdash_\Gamma \Diamond^u \bigvee_i \phi_i}$

$\frac{}{\Diamond^d \bigvee_i \phi_i \vdash_\Gamma \bigvee_i (\Diamond^d \phi_i)}$ $\frac{}{\bigvee_i (\Diamond^d \phi_i) \vdash_\Gamma \Diamond^d \bigvee_i \phi_i}$

μ_{10}) $\Box \bigwedge$ preservation $\frac{}{\Box^u \bigwedge_i \phi_i \vdash_\Gamma \bigwedge_i (\Box^u \phi_i)}$ $\frac{}{\bigwedge_i (\Box^u \phi_i) \vdash_\Gamma \Box^u \bigwedge_i \phi_i}$

$\frac{}{\Box^d \bigwedge_i \phi_i \vdash_\Gamma \bigwedge_i (\Box^d \phi_i)}$ $\frac{}{\bigwedge_i (\Box^d \phi_i) \vdash_\Gamma \Box^d \bigwedge_i \phi_i}$

μ_{11}) $\Diamond$ 1st mixed Frobenius $\frac{}{\Diamond^u (\phi \wedge \Diamond^d \psi) \vdash_\Gamma \Diamond^u (\Diamond^u \phi \wedge \Diamond^d \psi)}$ $\frac{}{\Diamond^u (\Diamond^u \phi \wedge \Diamond^d \psi) \vdash_\Gamma \Diamond^u (\phi \wedge \Diamond^d \psi)}$

μ_{12}) $\Diamond$ 2nd mixed Frobenius $\frac{}{\Diamond^d (\phi \wedge \Diamond^u \psi) \vdash_\Gamma \Diamond^d (\Diamond^d \phi \wedge \Diamond^u \psi)}$ $\frac{}{\Diamond^d (\Diamond^d \phi \wedge \Diamond^u \psi) \vdash_\Gamma \Diamond^d (\phi \wedge \Diamond^u \psi)}$

We interpret these operators using $\heartsuit_{\mathcal{X}}$ and $\triangle_{\mathcal{X}}$, respectively.

This satisfiability relation expresses different properties for p, depending on the fact that the interpretation of ϕ is a strict subobject of $\mathcal{X}$ or not. In fact, prefixes and prolongations of p are not the same in the two cases.

The rules presented in Table 2 reflect the given semantics: the inference rules labelled with $S4$ correspond to those characterizing the homonymous system of modal logic and to the fact that time relations are transitive in our model; rules μ_1 and μ_2 depend on reflexivity; rules μ_3 and μ_4 depend on functoriality. These rules, as well as those describing the interplay between modal operators and connectives, are directly established according to the categorical properties of their semantical counterparts. In particular, $\Box$'s, corresponding to right adjoints, preserve conjunction $\wedge$, whereas $\Diamond$'s, corresponding to left adjoints, preserve disjunction $\vee$. As for the quantifiers, we can say that existential quantifiers combined with $\Diamond$'s are again left adjoints, whereas universal quantifiers combined with $\Box$'s are again right adjoints.

The relationships between the strict and the non-strict case are illustrated via the diagrams in Figure 4. There the commutativity of the squares in the middle, namely the immediate validity of the equations $s_{\mathcal{X}} \circ i^u = i^s \circ s^u_{\mathcal{X}}$ and $s_{\mathcal{X}} \circ i^d = i^s \circ s^d_{\mathcal{X}}$, makes the external and the internal squares also commutative by the uniqueness of adjoints.

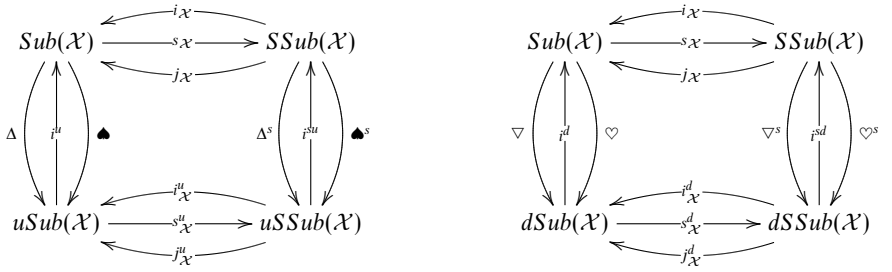

Fig. 4. The relationships between the modal doctrines.

4. Comparisons and applications

4.1. *The hyperdoctrinal approach*

Introduced by Lawvere (1969) and developed in Lawvere (1970), a hyperdoctrine is a cartesian closed category **T** of *types*, whose morphisms are called *terms*. For each type X, there is a cartesian closed category $\mathbf{P}(\mathbf{X})$ of *attributes* of type X, playing the role of generalised subobjects (whose aim is to interpret formulae), and morphisms between them (which are called *deductions* over X); for each term $f : X \to Y$, there is a functor $f^* : \mathbf{P}(\mathbf{Y}) \to \mathbf{P}(\mathbf{X})$, called *substitution*, and two functors Σ_f and Π_f, respectively left and right adjoint to substitution, called *existential* and *universal quantification along* f, respectively. A hyperdoctrine is a kind of indexed or fibered category. There are several variants of this concept, as it can be adapted to many logical frameworks (Pisani 2010; Streicher 2003). It is also well known that doctrines are able to express the logic of a categorical structure. Since we are only interested in studying different subobject structures of a given category (and, consequently, different logical systems that can be canonically associated with it), we have introduced a suitably simplified definition of doctrine. In this way, we agree with the approach in Pisani (2010), except for our assumption that all the objects in our doctrines are in the doctrine *Sub* of a category, which is at least Heyting, given by the actual subobjects. Hence, other properties that are often required, like Frobenius reciprocity and the comprehension scheme, are automatically satisfied.

It is also known that one can develop a reasonable modal/temporal logic using relational presheaves (Ghilardi and Meloni 1988) or, almost equivalently, quantale-enriched categories (Rosenthal 1993), instead of a topos. In fact, following Ghilardi and Meloni (1988), Rosenthal (1993) uses a doctrinal approach by considering attributes instead of subobjects to speak about properties. Here, we directly use a categorical structure, which is generated by a suitable semilattice and that is also strictly related to the enriched category theory, due to the isomorphism between L_B and $SymcatB$. Our approach confirms that a topos may be not the best structure from which to start to obtain a good modal/temporal logic: indeed, we do not need to add external attributes, but we can simply use subobjects already existing in the structure and subfamilies of them. Thus, we are able to isolate two families of temporal/modal operators directly

corresponding to the strict and non-strict cases, one of them living in a classical logic, the other one in a Heyting logic.

4.2. *Relationships with toposes*

The algebraic study of L_B shows that L_B fails to be a topos on its own (not every mono is regular), but contains a Boolean topos and intrinsically bears a modal/temporal structure. Actually, when we restrict to strict monos, we also have a subobject classifier. Let Ω be the object of L_B defined as follows: elements are all the pairs $(b, b') \in B \times B$, where $b' \leqslant b$. The extent of such an element is b, while agreement is defined as follows:

$$\alpha((b,b'),(c,c')) = \begin{cases} b \wedge c & \text{if } b' = c' \\ b' \wedge c' & \text{otherwise.} \end{cases}$$

The *true* function is the obvious immersion of the terminal object $\mathcal{B}$ into Ω sending b into (b, b). b' will represent the 'degree of membership' w.r.t. b. We define the characteristic function of a subobject $\mathcal{X}'$ of $\mathcal{X}$, via a strict mono m, in L_B as

$$\chi_m(x) = \begin{cases} (\iota_X(x), \iota_X(x)) & \text{if } x \in X' \\ (\iota_X(x), \bigvee_{x' \in X'} \alpha_X(x, x')) & \text{otherwise,} \end{cases}$$

where $\bigvee_{x' \in X'} \alpha_X(x, x')$ is well defined because it is bounded by $\iota_X(x)$.

To be a quasitopos, L_B should also be locally cartesian closed. The proof (or refutation) of such a property does not seem to be immediate, neither does it appear to be crucial for the present work; so, we leave it for future research.

4.3. *Modal systems associated with algebraic models of computing processes*

Temporal/modal logic is associated with algebraic models of process behaviours in different ways. As an important example, Kripke models are used as a semantics for Computational Tree Logic, CTL (and CTL*) (Huth and Ryan 2004), a very rich propositional modal logic, whose formulae are divided into path and state formulae but are interpreted only on states (the sets of states verifying them). Taking into account that Kripke models can be modelled by our trees, we first try to define CTL operators in our context. To this aim, we have to consider the case where the semilattice **B** corresponds to a discrete time model, e.g., when **B** is a free monoid, and compare a model of CTL with a model of our logic.

Let $\mathcal{M} = (S, \rightarrow, L)$ be a model of CTL, where S is a set of states, $\rightarrow$ represents the transition relation and L is the labelling function, assigning to every state the set of atomic formulae holding in that state. If the initial state is s_0, from $\mathcal{M}$ we can create $Unf(\mathcal{M})$, a tree whose paths are the initial paths in $\mathcal{M}$ starting from s_0, labelled with the sequences of states they go through and with maximal agreement, i.e., an object of the category L_B. One must be careful also with the notion of path: in our model, a path is always finite (since it is an initial path) and so it can be identified with its final state, so that we have a bijection between paths and states; the order between states is inherited

by the corresponding one on paths. An atomic formula will be interpreted as the subtree consisting of paths enjoying the formula at their terminal state.

The tree $Unf(\mathcal{M})$ is therefore deterministic and up-closed; thus, we will use the strict form of our operators (viz, $\spadesuit^s, \heartsuit^s, \triangle^s, \triangledown^s$) without indicating the superscript s. It is immediate to see that all usual operators of CTL, except for 'next', can be modelled through our operators.

Proposition 4.1. The four base operators $\diamond^u, \diamond^d, \Box^u, \Box^d$ can model CTL operators as follows:

— $AG\phi(p)$ is equivalent to $p \models_{\mathcal{X}} \Box^d \phi$,
— $EF\phi(p)$ is equivalent to $p \models_{\mathcal{X}} \diamond^u \phi$,
— $AF\phi(p)$ is equivalent to $p \models_{\heartsuit(\{p\})} \Box^d(\diamond^d\phi \vee \diamond^u\phi)$,
— $E(\phi U\psi)(p)$ is equivalent to $p \models_{\heartsuit(\{p\})} \diamond^u(\psi \wedge \Box^u\phi)$,

where $\heartsuit(\{p\})$ is the down closure, w.r.t. $\mathcal{X}$, of the subobject identified by $\{p\}$, i.e., $\langle\{p\}, p, p\rangle$.

Proof. The result is obtained by comparing the satisfiability conditions. □

The well-known duality between operators, i.e., $\Box\phi = \neg\diamond\neg\phi$ and $\diamond\phi = \neg\Box\neg\phi$ (Huth and Ryan 2004) is still valid since we are in the strict case, where negation is Boolean. We now introduce *next* operators in our context. We define the binary relation

$$Succ(x, x') \text{ iff } x' \leqslant x \text{ and there is } a \in A \text{ s.t. } \iota(x) = \iota(x')a$$

and from it we define, using the notation of Definition 3.3 and substituting $Succ(x, x')$ for $x' \leqslant x$, a subfunctor of $\spadesuit_{\mathcal{X}}$ and a subfunctor of $\triangledown_{\mathcal{X}}$.

Definition 4.1 (Next functors). We define the following *next functors:*

— $\spadesuit^{succ}_{\mathcal{X}} : Sub\mathcal{X} \to SSub\mathcal{X}$, where $\spadesuit^{succ}_{\mathcal{X}}\mathcal{X}' = \langle X'', \iota_{\spadesuit\mathcal{X}'}, \alpha_{\spadesuit\mathcal{X}'}\rangle$ is the strict subobject of $\spadesuit_{\mathcal{X}}\mathcal{X}'$ such that $X'' = \{x \in X \mid \exists x' \in X'[Succ(x', x)]\}$ and the other functions are inherited from $\spadesuit_{\mathcal{X}}\mathcal{X}'$.
— $\triangledown^{succ}_{\mathcal{X}} : Sub\mathcal{X} \to SSub\mathcal{X}$, where $\triangledown^{succ}_{\mathcal{X}}\mathcal{X}' = \langle X'', \iota_{\triangledown\mathcal{X}'}, \alpha_{\triangledown\mathcal{X}'}\rangle$ is the strict subobject of $\triangledown_{\mathcal{X}}X'$ such that $X'' = \{x \in X \mid \forall x' \in X[Succ(x', x) \Rightarrow x' \in X']\}$ and the other functions are inherited from $\triangledown_{\mathcal{X}}\mathcal{X}'$.

Note that, if we restrict to trees which are CTL models, these two functors are left and right adjoint to the functor that adds all the possible steps to a given subobject; as a consequence, we have another doctrine. Using these functors, we can define *next* operators in our logic in such a way that

— $p \models_{\mathcal{X}} X_{\diamond}\phi$ *iff* $\exists q[Succ(q, p) \wedge q \models_{\mathcal{X}} \phi]$, i.e. *iff* $p \in \spadesuit^{succ}_{\mathcal{X}}(|\phi|)$;
— $p \models_{\mathcal{X}} X_{\Box}\phi$ *iff* $\forall q[Succ(q, p) \Rightarrow q \models_{\mathcal{X}} \phi]$, i.e. *iff* $p \in \triangledown^{succ}_{\mathcal{X}}(|\phi|)$.

Proposition 4.2.

— $EX\phi(p)$ is equivalent to $p \models_{\mathcal{X}} X_{\diamond}\phi$,
— $AX\phi(p)$ is equivalent to $p \models_{\mathcal{X}} X_{\Box}\phi$.

Proof. The result is obtained by comparing the satisfiability conditions. □

Since the defined CTL operators are a base for all CTL operators, we have here that they are definable from our four base operators $\diamond^u, \diamond^d, \Box^u, \Box^d$ plus the next operators $X_\diamond$ and $X_\Box$.

Due to the substantial identification of states with their corresponding paths, our formulae are always concerned with paths/states. For this reason, while AG and EF can be simply translated (in fact their interpretation uses the same kind of quantifier for both paths and states), this is not the case with the other ones. Naturally, in order to mimic CTL/CTL*, we have to consider only the Boolean subcategory of L_B made up of strict up-closed objects, because CTL/CTL* are based on a Boolean logic. More accurate comparisons with CTL/CTL* and investigations about computational complexity of algorithms to verify properties needed for model checking are left to future work.

4.4. *Languages to model concurrency*

As a first extension to a non-deterministic context, let us now consider LTS associated with the Hennessy–Milner logic (HML) (Aceto et al. 2007; Hennessy and Milner 1985). This is a very simple (positive, untyped) propositional Boolean modal logic, whose formulae are interpreted on states thought of as their possible future behaviours. An LTS corresponds to an up-closed tree, where **B** is the free monoid A^* of elementary actions.

Note that formulae in HML are not interpreted as subobjects, so that a direct comparison with our approach is not possible. Nevertheless, we show how to define, in our context, the basic HML operators. Their introduction will be very similar to that of *next* operators. We define the binary relation

$$Succ^a(x, x') \text{ iff } x' \leqslant x \text{ and } \imath(x) = \imath(x') \cdot a$$

and from it we define, using the notation of Definition 3.3 and substituting $Succ^a(x, x')$ to $x' \leqslant x$, a subfunctor of $\spadesuit_{\mathcal{X}}$ and a subfunctor of $\triangledown_{\mathcal{X}}$.

Definition 4.2 (HML functors). We define

— $\spadesuit^a_{\mathcal{X}} : Sub\mathcal{X} \to SSub\mathcal{X}$, where $\spadesuit^a_{\mathcal{X}}\mathcal{X}' = \langle X'', \imath_{\spadesuit_{\mathcal{X}'}}, \alpha_{\spadesuit_{\mathcal{X}'}} \rangle$ is the strict subobject of $\spadesuit_{\mathcal{X}}\mathcal{X}'$ such that $X'' = \{x \in X \mid \exists x' \in X'[Succ^a(x', x)]\}$ and the other functions are inherited from $\spadesuit_{\mathcal{X}}\mathcal{X}'$.

— $\triangledown^a_{\mathcal{X}} : Sub\mathcal{X} \to SSub\mathcal{X}$, where $\triangledown^a_{\mathcal{X}}\mathcal{X}' = \langle X'', \imath_{\triangledown\mathcal{X}'}, \alpha_{\triangledown_{\mathcal{X}}\mathcal{X}'} \rangle$ is the strict subobject of $\triangledown_{\mathcal{X}}\mathcal{X}'$ such that $X'' = \{x \in X \mid \forall x' \in X[Succ^a(x', x) \Rightarrow x' \in X']\}$ and the other functions are inherited from $\triangledown_{\mathcal{X}}\mathcal{X}'$.

Note that these two functors are left and right adjoint to the functor that adds all the a-labelled possible steps to a given subobject; as a consequence, we have another doctrine. Using these functors, we can define $\langle a \rangle$ and $[a]$ operators in our logic in such a way that

— $p \models_{\mathcal{X}} \langle a \rangle \phi$ *iff* $\exists q[Succ^a(q, p) \wedge q \models_{\mathcal{X}} \phi]$, i.e. *iff* $p \in \spadesuit^a_{\mathcal{X}}(| \phi |)$;
— $p \models_{\mathcal{X}} [a]\phi$ *iff* $\forall q[Succ^a(q, p) \Rightarrow q \models_{\mathcal{X}} \phi]$, i.e. *iff* $p \in \triangledown^a_{\mathcal{X}}(| \phi |)$.

In the same line as HML, we find the coalgebraic approach to modal logic (see, e.g., Klin (2007)). In this research line, the idea is to consider expressiveness as the

capacity of defining subobjects (formulae), so that logically indistinguishable states are behaviourally equivalent.

In the case of the doctrinal approach adopted here, formulae do coincide with subobjects by definition, so it does not make sense to look for expressivity as the capacity of defining subobjects as before, but as the capacity of defining new operators, maybe in different situations; a representative sample is the continuous time case, hinted at in Section 2.2. There, using the same technique of changing the accessibility relation and applying it to the example of continuous trees, we can define other interesting modal operators. For example, let $x \preccurlyeq_t x'$ denote the relation which holds if and only if $x \preccurlyeq x'$ and, if $dom(\imath(x)) = [0, s)$, then $dom(\imath(x')) = [0, s + t')$, with $t \leqslant t'$. Then, the algebraic operators:

— $p \in \spadesuit^t_{\mathcal{X}}(|\ \phi\ |)$ *iff* $\exists q[(p \preccurlyeq_t q) \wedge q \in |\ \phi\ |]$
— $p \in \triangledown^t_{\mathcal{X}}(|\ \phi\ |)$ *iff* $\forall q[(p \preccurlyeq_t q) \wedge q \in |\ \phi\ |]$
— $p \in \heartsuit^t_{\mathcal{X}}(|\ \phi\ |)$ *iff* $\exists q[(q \preccurlyeq_t p) \wedge q \in |\ \phi\ |]$
— $p \in \triangle^t_{\mathcal{X}}(|\ \phi\ |)$ *iff* $\forall q[(q \preccurlyeq_t p) \wedge q \in |\ \phi\ |]$

have corresponding modal operators with the following meaning, respectively:

— After a time t, there will be a future when ϕ will be satisfied.
— After a time t, ϕ will be satisfied in all the futures.
— There was a time, past from at least t, when ϕ was satisfied.
— In all the times, past from at least t, ϕ was satisfied.

4.5. *Playing the Pacman game*

So far, we have not investigated the possible applications of the Heyting aspects of our logic. Now, we intend to illustrate the adequacy of the proposed logical framework to describe and reason about (possibly non-deterministic) processes, e.g., visual games.

Let us suppose we have a maze M. The system of all the paths (intended as sequences of moves) that allow a player to successfully go from the entrance position to the exit position, through a series of contiguous intermediate positions, can be modelled by a tree, $Tree(M)$, in our sense[‡]. In order to leave the maze, a player interacting with it has to produce a path identical to one in $Tree(M)$; in other words, the tree of the player's possible behaviours must synchronize with $Tree(M)$ at least along one path. $Tree(M)$ can be up-closed to represent also those (partial) paths that can be prolonged by a path in $Tree(M)$, obtaining $T(M)$.

Let us now suppose that the maze has an internal device that can non-deterministically change the position of the walls or introduce another obstacle which can unpredictably change an original successful path into an unsuccessful one. This is the case of the well-known Pacman game, where the original deterministic maze can be altered by the presence of a ghost which can interrupt some originally prolongable path. In our setting, this means that the originally deterministic situation, e.g., for a path starting with an s move (schematically illustrated on the right part of Figure 5 and already considered as

[‡] A path might include subsequences of moves which bring the player back to already visited positions.

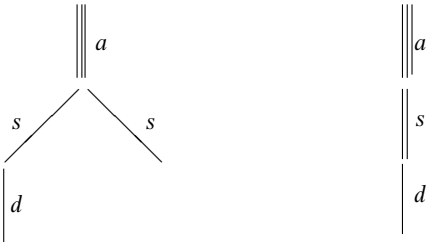

Fig. 5. A non-strict subobject and the corresponding strict subobject in the up-closure of $Tree(M)$.

up-closed), is changed in its non-deterministic subobject (illustrated on the left part of Figure 5) after the synchronization with the ghost.

The Pacman tries to synchronize with the maze as far as possible. Suppose it was instructed in such a way that, if its first move was an a, then its strategy would be successful, i.e., it will eventually perform the move d (eat a pellet) and stop (for the sake of simplicity, we suppose there is just one pellet). Figure 5 represents two possibilities for the interpretation of the formula ϕ stating that 'the first move is an a' in $T(M)$: the left side one is non-strict, the other one is strict. Then, the knowledge available to the Pacman (its initial instructions) would be represented by $\phi \Rightarrow \Diamond\delta$, where δ formalizes 'perform a move d and stop'.

So the Pacman starts with an a and is defeated, because its behaviour synchronized with the non-strict interpretation on the right-hand path. Then, it will complain: 'The instruction was incorrect; a correct one would have been $\neg\neg\phi \Rightarrow \Diamond\delta$'. In fact, the right-hand tree in Figure 5 is also the interpretation of $\neg\neg\phi$, while the left-hand tree is not, since the first one is the double complement of the second one. The double negation says that the success is guaranteed only under the extra hypothesis that things remain as in the original tree. In fact, only interpretations of doubly negated formulas are strict subobjects; here, we see our Heyting logic at work, since ϕ and $\neg\neg\phi$ are not equivalent.

5. Conclusions

The identification of a system of doctrines associated with the category L_B and of the relationships between them, not only provided us with a very rich language to speak about properties of a non-deterministic computation system, but also allowed us to analyse the impact of non-determinism on the logical structure of the proposed algebraic model.

Some word is in order about our view of non-determinism. As a matter of fact, our notion of non-determinism, which is related to the distinction between strict and non-strict monomorphisms, is always relative to a given type (i.e., an object in L_B), since a typed logic essentially deals with families of subobjects. As a consequence, a subobject can be more or less deterministic with respect to its type. Each type is of course deterministic with respect to itself, but sometimes it can be considered non-deterministic w.r.t. another type. An object in L_B is *intrinsically* deterministic if it can never be non-deterministic; this happens when its agreement w.r.t. to any other object, in particular the terminal one, cannot be increased, i.e., it is maximal.

In the case of strict subobjects of the terminal object, agreement is no longer relevant and deterministic agents are simply 'languages'. We could summarize the situation by saying that the distinction between determinism and non-determinism in L_B corresponds to the possibility of using just strict monomorphisms versus the need to consider all monomorphisms. This relates to the possibility of imposing a Boolean structure on the resulting category of subobjects, or having to resort to a weaker Heyting structure, where the relationship between Boolean and Heyting structure can be induced, as usual, by the $\neg\neg$ operator. Finally, depending on the structure imposed to the category, one has available the inference figures of classical or intuitionistic logic to derive inferences in the corresponding logics. In the context of L_B, one could say that the distinctions between inferences possible in the classical and intuitionistic logics, as derivable from the Boolean and Heyting doctrines, characterise the possibility of distinguish deterministic from non-deterministic processes.

Of course, the order (temporal) structure assumed for **B**, that allows the introduction of L_B, is strictly related with an assumption on non-determinism. In fact, when this structure becomes trivial, L_B collapses in more classical categories, such as Set or Set/B. However, this prevents the possibility of defining objects in L_B carrying a meaningful non-deterministic structure. Conversely, also the possibility of introducing several modal/temporal operators in a natural way is due, again, to the non-trivial, though poor, order structure of **B**.

Two main advantages derive from the approach followed in this paper: (1) by giving a definition of doctrine (temporal doctrine) relative to subobjects, the doctrinal approach adopted here allowed us to easily translate the 'conceptual' logic associated with a category in the sense of Lawvere (1970) into a 'formal' logic associated with the same category; (2) by combining the first-order intuitionistic logic and the modal logic, we obtained a potentially very flexible structure. As a price to pay, bisimulation is no longer characterisable through our modal version of the HML operators, nor was it meant to be, because our logic is essentially incomparable with HML.

We leave to future work a detailed analysis of the temporal logics we have obtained: in particular, we can study the comparisons with other modal/temporal logics already used in computer science, the computational complexity of its operators, or the possibilities arising from the use of the produced language in applications where non-determinism is relevant.

To conclude, we want to remark that our approach purposely does not make use of the fixed point technique to define modal operators; we leave for future work also the development of fixed points in our framework.

The authors would like to thank the anonymous referees: their work was very useful and stimulating.

Appendix A. Interpretation of a Sorted FirstOrder Language in a Category with Finite Limits

We here present some material from Johnstone (2002) used in this paper.

Definition A.1 (Σ-structure). Given a sorted first-order language $\mathcal{L}$ with signature Σ and a category $\mathcal{C}$ with finite limits, we define a Σ-structure in $\mathcal{C}$ by giving a mapping $|\cdot|$ satisfying the following:

— Each type A of $\mathcal{L}$ is mapped into an object $|A|$ of $\mathcal{C}$.
— Each function symbol f in Σ of type $A_1 \dots A_n \to A$ in $\mathcal{L}$ (written $f : A_1 \dots A_n \to A$) is mapped into an arrow $|f| : |A_1| \times \cdots \times |A_n| \to |A|$ of $\mathcal{C}$.
— Each predicate symbol R in Σ of type $A_1 \dots A_n$ in $\mathcal{L}$ (written $R : A_1 \dots A_n$) is mapped into a subobject $|R|$ of $|A_1| \times \dots \times |A_n|$, requiring that the basic predicate symbol $=_A$ of type AA should be mapped in the diagonal subobject of $|A| \times |A|$.

Definition A.2 (Interpretation of terms in a context). Given $\mathcal{L}$, $\mathcal{C}$, Σ and a mapping $|\cdot|$ defining a Σ-structure in $\mathcal{C}$ as in Definition A.1, we recursively define the *interpretation* $[\![\Gamma.t]\!]$ *of a term* t in a context $\Gamma = x_1 : A_1, \dots, x_n : A_n$ as follows:

1. If t is a variable, with $t = x_i$ for a unique i, then $[\![\Gamma.t]\!] = \pi_i$, where $\pi_i : |A_1| \times \cdots \times |A_n| \to |A_i|$.
2. If $t = f(t_1, \dots, t_m)$, then $[\![\Gamma.t]\!] = |f| \langle [\![\Gamma.t_1]\!], \dots, [\![\Gamma.t_m]\!] \rangle$.

Definition A.3 (Interpretation of formulae in a context). Given $\mathcal{L}$, $\mathcal{C}$, Σ and a mapping $|\cdot|$ defining a Σ-structure in $\mathcal{C}$ as in Definition A.1, we recursively define the interpretation $[\![\Gamma.\phi]\!]$ of a formula ϕ in a context $\Gamma = x_1 : A_1, \dots, x_n : A_n$ (where $\{x_1, \dots, x_n\}$ is the set of variables that appear free in ϕ) as follows:

1. If ϕ is a relation $R(t_1, \dots, t_m) : A_1 \dots A_m$, hence associated with a subobject of $|A_1| \times \cdots \times |A_m|$, then $[\![\Gamma.\phi]\!]$ is the pullback of this subobject along $\langle [\![\Gamma.t_1]\!], \dots, [\![\Gamma.t_m]\!] \rangle$.
2. If ϕ is $\top$, then $[\![\Gamma.\phi]\!]$ is the top element of $Sub(|A_1| \times \cdots \times |A_n|)$.
3. If ϕ is $\psi \wedge \chi$, then $[\![\Gamma.\phi]\!]$ is the pullback (intersection) of $[\![\Gamma.\psi]\!]$ and $[\![\Gamma.\chi]\!]$.
4. If ϕ is $\bot$, then $[\![\Gamma.\phi]\!]$ is the bottom element of $Sub(|A_1| \times \dots \times |A_n|)$.
5. If ϕ is $\psi \vee \chi$, then $[\![\Gamma.\phi]\!]$ is the union of $[\![\Gamma.\psi]\!]$ and $[\![\Gamma.\chi]\!]$.
6. If ϕ is $\exists y : A \psi$ and $\mathcal{C}$ is a regular category, $[\![\Gamma.\phi]\!]$ is the image of the composition of $[\![\Gamma, y : A.\phi]\!]$ with the projection $\pi : |A_1| \times \cdots \times |A_n| \times |A| \to |A_1| \times \cdots \times |A_n|$.
7. If ϕ is $\psi \Rightarrow \chi$ and $\mathcal{C}$ is a Heyting category, $[\![\Gamma.\phi]\!]$ is the implication between $[\![\Gamma.\psi]\!]$ and $[\![\Gamma.\chi]\!]$ in $Sub(|A_1| \times \cdots \times |A_n|)$.
8. If ϕ is $\neg\psi$ and $\mathcal{C}$ is a Heyting category, $[\![\Gamma.\phi]\!]$ is the negation of $[\![\Gamma.\psi]\!]$ in $Sub(|A_1| \times \cdots \times |A_n|)$.
9. If ϕ is $\forall y : A \psi$ and $\mathcal{C}$ is a Heyting category, $[\![\Gamma.\phi]\!]$ is $\Pi_\pi([\![\Gamma, y : A.\psi]\!])$, where $\pi : |A_1| \times \cdots \times |A_n| \times |A| \to |A_1| \times \cdots \times |A_n|$.
10. If ϕ is $\bigvee_{i \in I} \psi_i$ and $\mathcal{C}$ is a geometric category, $[\![\Gamma.\phi]\!]$ is the union over I of all $[\![\Gamma.\psi_i]\!]$ in $Sub(|A_1| \times \cdots \times |A_n|)$.
11. If ϕ is $\bigwedge_{i \in I} \psi_i$ and $\mathcal{C}$ has arbitrary intersection of subobjects, $[\![\Gamma.\phi]\!]$ is the intersection over I of all $[\![\Gamma.\psi_i]\!]$ in $Sub(|A_1| \times \cdots \times |A_n|)$.

Appendix B. Definition of Temporal Doctrine

We rephrase here the definition of temporal doctrine by Pisani (2010) according to our notation.

Definition B.1 (Temporal doctrine). A temporal doctrine

$$i^u : u\mathcal{D} \to \mathcal{D} \leftarrow d\mathcal{D} : i^d$$

consists of two c-indexed functors (where c is an object of $\mathbf{C}$) with the same codomain, viz. the hyperdoctrine (c-indexed category) $\mathcal{D}$, satisfying the properties listed below.

1. The indexing category $\mathbf{C}$ has a terminal object.
2. The categories in $\mathcal{D}(c)$ are cartesian closed.
3. The substitution functors $f^* : \mathcal{D}(e) \to \mathcal{D}(c)$ have both left and right adjoints.
4. The functors $i^u_c : u\mathcal{D}(c) \to \mathcal{D}(c)$ and $i^d_c : d\mathcal{D}(c) \to \mathcal{D}(c)$ have both left and right adjoints:
$$\diamond^u_c \dashv i^u_c \dashv \Box^u_c \qquad\qquad \diamond^d_c \dashv i^d_c \dashv \Box^d_c.$$
5. The doctrine $\mathcal{D}$ satisfies the comprehension axiom (Lawvere 1970): the canonical functors $\sigma_c : \mathbf{C}/c \to \mathcal{D}(c)$ (sending $f : d \to c$ to $\Sigma_f 1_d$) have right adjoints: $\sigma_c \dashv \tau_c : \mathcal{D}(c) \to \mathbf{C}/c$.
6. The functors i^u_c and i^d_c are fully faithful:
$$\diamond^u_c i^u_c \simeq id; \Box^u_c i^u_c \simeq id \qquad\qquad \diamond^d_c i^d_c \simeq id; \Box^d_c i^d_c \simeq id.$$
7. The doctrine $\mathcal{D}$ satisfies the Frobenius law: $\Sigma_f P \times_e Q \simeq \Sigma_f (P \times_c f^* Q)$, for any $f : c \to e$ (naturally in $P \in \mathcal{D}(c)$ and $Q \in \mathcal{D}(e)$).
8. Left adjoints satisfy the mixed Frobenius law, i.e., the units of adjunctions induce isomorphisms:
$$\diamond^u_c(P \times_c i^d_c N) \simeq \diamond^u_c(i^u_c \diamond^u_c P \times_c i^d_c N)$$
$$\diamond^d_c(P \times_c i^u_c M) \simeq \diamond^d_c(i^d_c \diamond^d_c P \times_c i^u_c M)$$
natural in $P \in \mathcal{D}(c)$, $M \in u\mathcal{D}(c)$ and $N \in d\mathcal{D}(c)$.
9. The projections $\pi^u 1 : u\mathcal{D}1 \times_{\mathcal{D}1} d\mathcal{D}1 \to u\mathcal{D}1$ and $\pi^d 1 : u\mathcal{D}1 \times_{\mathcal{D}1} d\mathcal{D}1 \to d\mathcal{D}1$ are isomorphisms.
10. The comprehension functors $\kappa_c : \mathcal{D}(c) \to \mathbf{C}/c$ are fully faithful.

The first five axioms require the existence of some adjoint functors; the next five axioms impose some exactness condition on these functors. In our case, where all the doctrines involved are required to be subfunctors of *Sub*, some axioms are automatically satisfied (namely 5, 7, 9 and 10) so that we can assume the simplified Definition 3.1.

References

Aceto, L., Ingólfsdóttir, A., Larsen, K.G. and Srba, J. (2007). *Reactive Systems: Modelling, Specification and Verification*, Cambridge University Press.

Borceux, F. (1994). *Handbook of Categorical Algebra: Volume 1, Basic Category Theory*, Encyclopedia of Mathematics and its Applications, Cambridge University Press. Available at `http://opac.inria.fr/record=b1126837`

Bottoni, P., Labella, A. and Kasangian, S. (2012). Spatial and temporal aspects in visual interaction. *Journal of Visual Languages and Computing* **23**(2) 91–102. Special issue dedicated to Prof. Piero Mussio. Available at `http://www.sciencedirect.com/science/article/pii/S1045926X11000772`

Braüner, T. and Ghilardi, S. (2007). First-order modal logic. In: Patrick Blackburn, J.V.B. and Wolter, F. (eds.) *Handbook of Modal Logic*, Studies in Logic and Practical Reasoning, vol. 3, Elsevier, 549–620. http://www.sciencedirect.com/science/article/pii/S1570246407800127

Cardelli, L. (1982). Real time agents. In: *Proceedings of the 9th Colloquium on Automata, Languages and Programming* 94–106. Available at http://dl.acm.org/citation.cfm?id=646236.682864

De Nicola, R., Gorla, D. and Labella, A. (2010). Tree-functors, determinacy and bisimulations, *Mathematical Structures in Computer Science* **20** 319–358. Available at http://journals.cambridge.org/article_S0960129509990272

Freyd, P. and Scedrov, A. (1990). *Categories, Allegories*, Mathematical Library, vol. 39, North-Holland.

Ghezzi, R. (2012). Enriched categories and presheaves: Their interconnections, generators and logics, Master's thesis, Università degli studi di Milano.

Ghilardi, S. and Meloni, G.C. (1988). Modal and tense predicate logic: Models in presheaves and categorical conceptualization, In: Borceux, F. (ed.) *Categorical Algebra and its Applications,* Springer, 130–142. Available at http://dx.doi.org/10.1007/BFb0081355

Hennessy, M. and Milner, R. (1985). Algebraic laws for nondeterminism and concurrency. *Journal of the ACM* **32**(1) 137–161. http://doi.acm.org/10.1145/2455.2460

Huth, M. and Ryan, M. (2004). *Logic in Computer Science: Modelling and Reasoning About Systems*, Cambridge University Press.

Johnstone, P. T. (2002). *Sketches of An Elephant: A Topos Theory Compendium*, vol. 1, Oxford Logic Guides, Clarendon Press. Available at http://opac.inria.fr/record=b1107183

Kasangian, S. and Labella, A. (1999). Observational trees as models for concurrency. *Mathematical Structures in Computer Science* **9**(6) 687–718. Available at http://dx.doi.org/10.1017/S0960129599002935

Klin, B. (2007). Coalgebraic modal logic beyond sets. *Electronic Notes in Theoretical Computer Science* **173** 177–201.

Lawvere, B. (1970). Equality in hyperdoctrines and the comprehension schema as an adjoint functor. In: Heller, A. (ed.) *Applications of Categorical Algebra*, Proceedings of Symposia in Pure Mathematics, vol. 17, American Mathematical Society, 1–14.

Lawvere, F. W. (1969). Adjointness in foundations. *Dialectica* **23**(3–4) 281–296. Available at http://dx.doi.org/10.1111/j.1746-8361.1969.tb01194.x

Milner, R. (1989). *Communication and Concurrency*, Prentice-Hall.

Pisani, C. (2010). A logic for categories. *Theory and Applications of Categories* **24** 394–417.

Pitts, A. M. (2000). Categorical logic. *Handbook of Logic in Computer Science*, Oxford University Press, 39–123. Available at http://dl.acm.org/citation.cfm?id=373919.373928

Rosenthal, K. I. (1993). 'A note on categories enriched in quantaloids and modal and temporal logic, *Cahiers de Topologie et Géométrie Différentielle Catégoriques* **34**(4) 267–277. Available at http://eudml.org/doc/91529

Streicher, T. (2003). Categorical models of constructive logic. Available at http://www.mathematik.tu-darmstadt.de/∼streicher/cmcl.pdf.

Walters, R. F. C. (1981). Sheaves and cauchy-complete categories. *Cahiers de Topologie et Géométrie Différentielle Catégoriques* **22**(3) 283–286. Available at http://eudml.org/doc/91273