Formal Relationships Between Geometrical and Classical Models for Concurrency

Eric Goubault and Samuel Mimram

CEA LIST*

Abstract. A wide variety of models for concurrent programs has been proposed during the past decades, each one focusing on various aspects of computations: trace equivalence, causality between events, conflicts and schedules due to resource accesses, etc. More recently, models with a geometrical flavor have been introduced, based on the notion of cubical set. These models are very rich and expressive since they can represent commutation between any number of events, thus generalizing the principle of *true concurrency*. While they seem to be very promising – because they make possible the use of techniques from algebraic topology in order to study concurrent computations – they have not yet been precisely related to the previous models, and the purpose of this paper is to fill this gap. In particular, we describe an adjunction between Petri nets and cubical sets which extends the previously known adjunction between Petri nets and asynchronous transition systems by Nielsen and Winskel.[1]

1 Introduction

There is a great variety of models for concurrency, which were introduced in the last decades: transition systems (with independence), asynchronous automata, event structure, Petri nets, etc. Each of these models focuses on modeling a particular aspect of computations, and even though their nature are very different, they are tightly related to each other as witnessed in [32]. More recently, models inspired by ideas coming from geometry were introduced, such as *cubical sets* (also sometimes called *higher dimensional automata* or HDA [21,13]) or local po-spaces [8]. Since then, they have not been systematically and formally linked with the other models, such as transition systems, even though cubical sets contain a notion of generalized transition in their very definition. The idea of relating these by adjunctions in the style of Winskel et al. [32] with operational models for concurrency dates back to [11], but this was done only between fairly restricted categories. In this paper, we greatly improve previous work by extending it to the full categories of transition systems (operational model of "interleaving" concurrency) and of transition systems with independence (operational model of "true" concurrency). The main motivation underlying this work is that, by relating these models, we can compare the semantics of concurrent languages given in different formalisms. Moreover, it is hoped that specific methods for statically analyzing concurrent programs in one model (such as deadlock detection algorithms for cubical sets [7], invariant generation on Petri nets [23], state-space reduction techniques such as

* CEA LIST, Laboratory for the Modelling and Analysis of Interacting Systems, Point Courrier 94, 91191 Gif-sur-Yvette, France.

1 This work has been supported by the PANDA ("Parallel and Distributed Analysis", ANR-09-BLAN-0169) French ANR project

sleep sets and persistent sets in Mazurkiewicz traces [10], or stubborn sets in Petri nets [28]) can be reused in the other, giving some nice cross-fertilization.

This paper constitutes a first step towards formally linking geometric models with other models for concurrency. The links might appear as intuitive, but the formal step we are making underlines subtle differences between the models, and unravels interesting phenomena (besides being necessary for being able to relate semantics given in different styles) such as the fact that persistent set types of methods for tackling the state-space explosion problem can be seen as searching for retracts of the state space, in the algebraic topological sense. We end this article by making some hypotheses on further relationships, with event structures and Petri nets in particular.

Related work. In this paper, we extend Winskel's results of [32] which includes adjunctions between transition systems, event structures, trace languages, asynchronous transition systems and Petri nets. A first step towards comparing higher-dimensional automata (a form of geometric semantics we are considering here), Petri nets, and event structures is reported in [30]. First steps towards the comparison between cubical sets (another form of geometric semantics) and transition systems, as well as transition systems with independence were described in [12] but never formally published.

We describe right adjoint functors from the categories of transition systems, asynchronous transition systems, Petri nets and prime event structures of [32], to HDA. By general theorems, these functors transport limits onto limits, hence preserve classical parallel semantics based on pullbacks, by synchronized products [1], as the ones in transition systems or the ones of [32].

Cubical sets that we take as the primary model for geometric semantics here, have appeared in numerous previous works, in algebraic topology in particular, see [26,3]. A monoidal presentation can also be found in [15]. The basics of "directed algebraic topology" that is at the basis of the mathematics involved in the geometric semantics we use here can be found in [14].

Contents of the paper. We begin by recalling some well-known models for concurrent computations (transition systems, asynchronous automata, event structures and Petri nets) in Section 2. We then introduce the geometric model provided by cubical sets in Section 3 and relate them to the previous models by defining adjunctions in Section 4. HDA naturally "contain" transition systems (resp. asynchronous transition systems), which just encode the non-deterministic (resp. and pairwise independence) information. Event structures are also shown to be more abstract than HDA: they impose binary conflict relations and conjunctive dependencies (an event cannot depend on a disjunction of two events), and they do not distinguish different occurrences of the same event. Petri nets have a built-in notion of degree of parallelism, as is the case of HDA (given by cell dimension) but impose specific constraints on dynamics. We finally conclude on future works in Section 5.

2 Traditional models for concurrency

2.1 Transition systems

Transition systems are one of the oldest semantic models, both for sequential and concurrent systems, in which computations are modeled as the sequence of interactions that they can have with their environment. There is a convenient categorical treatment of this model, that we use in the sequel, taken from [32].

Definition 1. *A* transition system *is a quadruple* $(S, i, E, Tran)$ *where*
- S *is a set of* states *with* initial state i
- E *is a set of* events*, and*
- $Tran \subseteq S \times E \times S$ *is the* transition *relation*

In other words, a transition system is a graph with a distinguished vertex. Transition systems are made into a category by defining morphisms to be some kind of simulation (for then being able to discuss about properties modulo weak or strong bisimulation, see [16]). The idea is that a transition system T_1 simulates a transition system T_0 if as soon as T_0 can fire some action a in some context, T_1 can fire a as well in some related context. A morphism $f : T_0 \to T_1$ defines the way states and transitions of T_0 are related to states and transitions of T_1 making transition systems into a category **TS**.

Definition 2. *Let* $T_0 = (S_0, i_0, E_0, Tran_0)$ *and* $T_1 = (S_1, i_1, E_1, Tran_1)$ *be two transition systems. A* partial morphism $f : T_0 \to T_1$ *is a pair* $f = (\sigma, \tau)$ *where* $\sigma : S_0 \to S_1$ *is a function and* $\tau : E_0 \to E_1$ *is a* partial *function such that*
- $\sigma(i_0) = i_1$,
- $(s, e, s') \in Tran_0$ *and* $\tau(e)$ *is defined implies* $(\sigma(s), \tau(e), \sigma(s')) \in Tran_1$. *Otherwise, if* $\tau(e)$ *is undefined then* $\sigma(s) = \sigma(s')$.

Idle transitions. As in [32], we can restrict to total morphisms, i.e. the ones for which τ is a total function, by suitably completing transition systems. Partial morphisms can then be recovered by adding "idle" transitions to the systems. This is closely related to the fact that the category $\mathbf{Set}'$ of sets and partial functions can be constructed as the Kleisli category [17] associated to the monad T on **Set** which associates to every set E the free pointed set $T(E) = E \uplus \{*\}$ (a pointed set is a set together with a distinguished element $*$). We will often use the fact that a partial function $A \to B$ can be seen as a pointed function $A \uplus \{*\} \to B \uplus \{*\}$ (that is a function sending $*$ to $*$), the element $*$ meaning "undefined".

An idle transition is a transition $*$ which goes from a state s to the same state s. Consider the following completion $T_* = (S_*, i_*, E_*, Tran_*)$ of a transition system $T = (S, i, E, Tran)$, by setting $S_* = S$, $i_* = i$, $E_* = E \uplus \{*\}$ and $Tran_* = Tran \uplus \{(s, *, s) \mid s \in S\}$. Now, by the preceding remarks a total morphism (σ, τ) from $(T_0)_*$ to $(T_1)_*$ such that $\tau(*) = *$ is the same as a partial morphism from T_0 to T_1. Again, the operation $(-)_*$ induces a monad on the category **sTS** of transition systems and total morphisms, and the category **TS** can be recovered as the Kleisli category associated to this monad.

Labeled transition systems. A *labeled transition system* consists of a transition system $T = (S, i, E, Tran)$ together with a set L of *labels*, a function $\ell : E \to L$ and a morphism $(\sigma, \tau, \lambda) : (T_1, L_1, \ell_1) \to (T_2, L_2, \ell_2)$ between labeled transition systems consists of a morphism $(\sigma, \tau) : T_1 \to T_2$ between the underlying transition systems together with a partial function $\lambda : L_1 \to L_2$ such that $\ell_2 \circ \tau = \lambda \circ \ell_1$. We write **LTS** for the category of labeled transition systems.

2.2 Asynchronous automata

Asynchronous automata are a nice generalization of both transition systems and Mazurkiewicz traces, and have influenced a lot of other models for concurrency, such as transition systems with independence (or asynchronous transition systems). They have been independently introduced in [27] and [2]. The idea is to decorate transition systems with an *independence* relation between actions that will allow us to distinguish between true-concurrency and mutual exclusion (or non-determinism) of two actions. We actually use a slight modification for our purposes, due to [5], called *automaton with concurrency relations*:

Definition 3. *An* automaton with concurrency relations $(S, i, E, Tran, I)$ *is a quintuple where*
- $(S, i, E, Tran)$ *is a transition system,*
- $Tran$ *is such that whenever* (s, a, s'), $(s, a, s'') \in Tran$, *then* $s = s''$,
- $I = (I_s)_{s \in S}$ *is a family of irreflexive, symmetric binary relations* I_s *on* E *such that whenever* $a_1 \, I_s \, a_2$ *(with* $a_1, a_2 \in E$*), there exist transitions* (s, a_1, s_1), (s, a_2, s_2), (s_1, a_2, r) *and* (s_2, a_1, r) *in* $Tran$.

A morphism of automata with concurrency relations consists of a morphism (σ, τ) between the underlying transition systems such that $a \, I_s \, b$ implies $\tau(a) \, I'_{\sigma(s)} \, \tau(b)$ whenever $\tau(a)$ and $\tau(b)$ are both defined. This makes automata with concurrency relations into a category, written **ACR**. Similarly to Section 2.1, we can equivalently consider **ACR** to be built using $*$-transitions and total morphisms.

2.3 Event structures

Event structures were introduced in [19,31] in order to abstract away from the precise places and times at which events occur in distributed systems. The idea is to focus on the notion of event and the causal ordering between them. We recall below the definition of (unlabeled prime) event structures.

Definition 4. *An* event structure $(E, \leqslant, \#)$ *consists of a poset* $(E, \leqslant)$ *of* events, *the partial order relation expressing* causal dependency, *together with a symmetric irreflexive relation* $\#$ *called* incompatibility *satisfying*
- *finite causes: for every event* e, *the set* $\{ \, e' \mid e' \leqslant e \, \}$ *is finite,*
- *hereditary incompatibility: for every events* e, e' *and* e'', $e \# e'$ *and* $e' \leqslant e''$ *implies* $e \# e''$.

We write **ES** for the category of event structures, a morphism between two event structures $(E, \leqslant, \#)$ and $(E', \leqslant', \#')$ consisting of a partial function $f : E \to E'$ such that if $f(e)$ is defined then $\{ \, e' \mid e' \leqslant f(e) \, \} \subseteq f(\{ \, e'' \mid e'' \leqslant e \, \})$, and if $f(e_0)$ and $f(e_1)$ are both defined and either $f(e_0) \#' f(e_1)$ or $f(e_0) = f(e_1)$ then either $e_0 \# e_1$ or $e_0 = e_1$.

2.4 Petri nets

Petri nets are a well-known model of parallel computation, generalizing transition systems by using a built-in notion of resource. This allows for deriving a notion of independence of events, which is much more general than the independence relation of asynchronous transition systems. They are numerous variants of Petri nets since they were introduced in [20], and we choose the definition used by Winskel and Nielsen in [32], since this is well-suited for formal comparisons with other models for concurrency:

Definition 5. *A* Petri net N *is an uple* $(P, M_0, E, \mathrm{pre}, \mathrm{post})$ *where*

- P *is a set of* places*,*
- $M_0 \in \mathbb{N}^P$ *is the* initial marking*,*
- E *is a set of* events*,*
- $\mathrm{pre} : E \to \mathbb{N}^P$ *and* $\mathrm{post} : E \to \mathbb{N}^P$ *are the* precondition *and* postcondition *functions.*

When there is no ambiguity, given an event e of a Petri net N, we often write ${}^\bullet e$ for $\mathrm{pre}(e)$ and $e^\bullet$ for $\mathrm{post}(e)$. A *marking* M is a function in $\mathbb{N}^P$, which associates to every place the number of resources (or tokens) that it contains. The sum $M_1 + M_2$ of two markings M_1 and M_2 is their pointwise sum. An event e induces a *transition* between two markings M_1 and M_2, that we write $M_1 \xrightarrow{e} M_2$, whenever there exists a marking M such that $M_1 = M + {}^\bullet e$ and $M_2 = M + e^\bullet$.

A *morphism of Petri nets* $(\varphi, \psi) : N \to N'$ where $N = (P, M_0, E, \mathrm{pre}, \mathrm{post})$ and $N' = (P', M'_0, E', \mathrm{pre}', \mathrm{post}')$ are two nets consists of a function $\varphi : P' \to P$ and a partial function $\psi : E \to E'$ such that for every place $p \in P'$ and event $e \in E$, $M'_0 = M_0 \circ \varphi$, ${}^\bullet\psi(e) = {}^\bullet e \circ \varphi$ and $\psi(e)^\bullet = e^\bullet \circ \varphi$. We write **PNet** for the category of Petri nets. Notice that the partial function $\varphi : P' \to P$ goes "backwards". This might seem a bit awkward at first sight and we explain why this is the "right" notion of morphism in Remark 1.

3 Geometric models for concurrency

Precubical sets can be thought as some sort of generalized transition system with higher-dimensional transitions. Similarly to transition systems there is a corresponding notion with "idle transitions", called *cubical sets*. These classical objects in combinatorial algebraic topology, see for instance [26], have been used as an alternative *truly concurrent* model for concurrency, in particular since the seminal paper [21] and [29]. More recently they have been used in [7] and [8] for deriving new and interesting deadlock detection algorithms. More algorithms have been designed since then, see for instance [22], [6] and [9].

3.1 Cubical sets

The cubical category. The *cubical category* $\square$ is the free category on the graph whose objects are natural integers $n \in \mathbb{N}$ and containing, for every integers i and n, such that $0 \leqslant i \leqslant n$, and for every $\alpha \in \{-, +\}$ arrows

$$\varepsilon^\alpha_{i,n} : n \to n+1 \quad \text{and} \quad \eta_{i,n} : n+1 \to n$$

(we sometimes omit the index n in the following) quotiented by the relations

$$\varepsilon^\alpha_j \varepsilon^\beta_i = \varepsilon^\beta_i \varepsilon^\alpha_{j+1} \qquad\qquad \eta_i \eta_j = \eta_{j+1}\eta_i \tag{1}$$

with $i \leqslant j$ and $\alpha, \beta \in \{-, +\}$ and, for every $\alpha \in \{-, +\}$,

$$\eta_j \varepsilon^\alpha_i = \begin{cases} \varepsilon^\alpha_i \eta_{j-1} & \text{if } i < j \\ \mathrm{id} & \text{if } i = j \\ \varepsilon^\alpha_{i-1}\eta_j & \text{if } i > j. \end{cases} \tag{2}$$

A monoidal definition of the cubical category. The resulting category $\square$ can be equipped with a very natural monoidal structure as follows. Given integers m, n, i such that $i \leqslant n$ and $\alpha \in \{-,+\}$, we write

$$\varepsilon^{\alpha}_{i,n} \otimes m = \varepsilon_{i,n+m} \quad m \otimes \varepsilon^{\alpha}_{i,n} = \varepsilon^{\alpha}_{i+m,n+m} \quad \eta_{i,n} \otimes m = \eta_{i,n+m} \quad m \otimes \eta_{i,n} = \eta_{i+m,n+m}$$

and extend this operation as a morphism, i.e. $(g \circ f) \otimes m = (g \otimes m) \circ (f \otimes m)$ and $m \otimes (g \circ f) = (m \otimes g) \circ (m \otimes f)$. It can easily be checked that $(\square, \otimes, 0)$ is a strict monoidal category, with the tensor product of two morphisms $f_1 : m_1 \to n_1$ and $f_2 : m_2 \to n_2$ being defined as $f_1 \otimes f_2 = (n_2 \otimes f_1) \circ (f_2 \otimes m_1)$.

We write $\varepsilon^{\alpha} = \varepsilon^{\alpha}_{0,0}$, with $\alpha \in \{-,+\}$, and $\eta = \eta_{0,0}$. These morphisms satisfy the equalities $\varepsilon^{-} \circ \eta = \mathrm{id}_0 = \varepsilon^{+} \circ \eta$, and moreover the morphisms $\varepsilon^{\alpha}_{i,n}$ and $\eta_{i,n}$ can be recovered from those morphisms by $\varepsilon^{\alpha}_{i,n} = i \otimes \varepsilon^{\alpha} \otimes (n-i)$ and $\eta_{i,n} = i \otimes \eta \otimes (n-i)$. All this suggests that we actually need much less data than in the traditional definition of the cubical category if we take the monoidal structure into account when generating the category, following the principle formalized by Burroni with polygraphs [4]. In fact, it can be shown [15] that

Property 1. The cubical category is the free monoidal category containing a co-cubical object.

Definition 6. *A* cubical object $(C, \varepsilon^{-}, \varepsilon^{+}, \eta)$ *in a monoidal category* $(\mathcal{C}, \otimes, I)$ *is an object* C *together with three morphisms* $\eta : I \to C$, $\varepsilon^{-} : C \to I$ *and* $\varepsilon^{+} : C \to I$ *such that* $\varepsilon^{-} \circ \eta = \mathrm{id}_I = \varepsilon^{+} \circ \eta$ *where* I *is the unit of the monoidal category. A* morphism f *between two cubical objects* $(C_1, \varepsilon^{-}_1, \varepsilon^{+}_1, \eta_1)$ *and* $(C_2, \varepsilon^{-}_2, \varepsilon^{+}_2, \eta_2)$ *is a morphism* $f : C_1 \to C_2$ *such that* $f \circ \eta_1 = \eta_2$, $\varepsilon^{-}_2 \circ f = \varepsilon^{-}_1$ *and* $\varepsilon^{+}_2 \circ f = \varepsilon^{+}_1$. *Dually, a* co-cubical object $(C, \varepsilon^{-}, \varepsilon^{+}, \eta)$ *in* $\mathcal{C}$ *is a cubical object in* $\mathcal{C}^{\mathrm{op}}$. *The* cubical category $\square$ *is the free monoidal category containing a co-cubical object.*

In other words, given a monoidal category $\mathcal{C}$, a cubical object in $\mathcal{C}$ is the same as a monoidal functor $\square^{\mathrm{op}} \to \mathcal{C}$. More precisely, the category of cubical objects in $\mathcal{C}$ is equivalent to category $\mathcal{C}^{\square^{\mathrm{op}}}$ of (strict) monoidal functors and monoidal natural transformations. This definition of cubical sets has been known for quite some time but no concrete application of it has been done. Interestingly, we show here that it can be used to concisely define some cubical sets (see in particular Section 3.3).

Cubical sets. A *cubical set* C is a presheaf on the cubical category, that is a functor $C : \square^{\mathrm{op}} \to \mathbf{Set}$, and we write **CSet** for the category of cubical sets and natural transformations between them. Concretely, a cubical set thus consists of a family $(C(n))_{n \in \mathbb{N}}$ of sets, whose elements are called *n-cells*, together with for every integers n and i, such that $0 \leqslant i \leqslant n$, maps $\partial^{-}_i, \partial^{+}_i : n+1 \to n$ and $\iota_i : n \to n+1$ satisfying axioms dual to those given for cubical sets (1) and (2). A morphism $\kappa : C \to C'$ between two cubical sets C and C' consists of a family $(\kappa_n : C(n) \to C'(n))_{n \in \mathbb{N}}$ of functions which are natural (i.e. $\kappa_{n+1} \circ \partial^{-}_i = \partial^{-}_i \circ \kappa_n$, etc.). We sometimes write $\partial^{-}_i = C(\varepsilon^{-}_i)$, $\partial^{+}_i = C(\varepsilon^{+}_i)$ for the *source* and *target* maps, and $\iota_i = C(\eta_i)$ for the *degeneracy* maps. The 0-*source* (resp. 0-*target*) of an n-cell $x \in C(n)$ is the 0-cell $\partial^{-}_0 \ldots \partial^{-}_0(x)$ (resp. $\partial^{+}_0 \ldots \partial^{+}_0(x)$).

Precubical sets. The *precubical category* $\boxdot$ is the full monoidal subcategory of $\square$ generated by the two morphisms η^{-} and η^{+}. A presheaf on the precubical category is called a *precubical set* and we write **PCSet** for the corresponding category. The precubical category can also be defined as the free monoidal category containing a co-precubical object $(C, \varepsilon^-, \varepsilon^+)$, that is an object C together with two arrows $\varepsilon^-, \varepsilon^+ : C \to I$.

Truncated cubical sets. Given an integer n, we write $\square_n$ for the full subcategory of $\square$ whose objects are the integers $k \leqslant n$. An *n-dimensional cubical set* is a presheaf on $\square_n$ and we write $\mathbf{CSet}_n$ for the category of n-dimensional cubical sets. The inclusion functor $\square_n \to \square$ induces by precomposition a functor $U_n : \mathbf{CSet} \to \mathbf{CSet}_n$ called the *n-truncation functor* (see Section 3.6).

3.2 Symmetric cubical sets

We sometimes need more structure on cubical sets in order to formally express the fact that the cells of dimension $n \geqslant 2$ in cubical sets arising as models for concurrent processes are essentially not directed (we explain this in details in Section 4.2). The *symmetric cubical category* $\square_S$ is the free symmetric monoidal category containing a co-cubical object. The presheaves on this category are called *symmetric cubical sets* and they form a category **SCSet**. The category $\square_S$ can also be described as the free monoidal category containing a symmetric co-cubical object $(C, \varepsilon^-, \varepsilon^+, \eta, \gamma)$, which is a co-cubical object $(C, \varepsilon^-, \varepsilon^+, \eta)$ together with a morphism $\gamma : C \otimes C \to C \otimes C$ such that

$$(\gamma \otimes C) \circ (C \otimes \gamma) \circ (\gamma \otimes C) = (C \otimes \gamma) \circ (\gamma \otimes C) \circ (C \otimes \gamma) \quad \text{and} \quad \gamma \circ \gamma = \gamma \quad (3)$$

and

$$\begin{array}{lll} \gamma \circ (\varepsilon^- \otimes C) = C \otimes \varepsilon^- & \gamma \circ (\varepsilon^+ \otimes C) = C \otimes \varepsilon^+ & (\eta \otimes C) \circ \gamma = C \otimes \eta \\ \gamma \circ (C \otimes \varepsilon^-) = \varepsilon^- \otimes C & \gamma \circ (C \otimes \varepsilon^+) = \varepsilon^- \otimes C & (C \otimes \eta) \circ \gamma = \eta \otimes C \end{array}$$

More details can be found in [15]. From this, the notion of symmetric cubical set can easily be reformulated as a cubical set C together with an action of the symmetric group Σ_n on $C(n)$ which satisfies suitable coherence axioms. Namely, any symmetry $\sigma : n \to n$ (i.e. a bijection on a set with n elements) can be decomposed as a product of transposition and can therefore be seen as a morphism in $\square_S$ by sending the transposition $\sigma_i : n \to n$, which exchanges the i-th and $(i+1)$-th element, to the morphism $i \otimes \gamma \otimes (n - i - 2)$. The axioms (3) as well as the axioms of monoidal categories ensure that this operation is well defined. In the following, we will implicitly see a bijection as a morphism in the category $\square_S$.

Given a symmetric monoidal category $\mathcal{C}$ (such as **Set** with cartesian product), any cubical object of the underlying monoidal category of $\mathcal{C}$ can be canonically equipped with a structure of symmetric cubical set, the morphism γ being given by the symmetry of the category.

Given an integer n, we write $(\square_S)_n$ for the full subcategory of $\square_S$ whose objects are integers $k \leqslant n$ and $\mathbf{SCSet}_n$ for the category of presheaves on $(\square_S)_n$, whose objects are called *n-dimensional symmetric cubical sets.*

3.3 Labeled cubical sets

Suppose that we are given a set L of labels. The category $(\mathbf{Set}, \times, 1)$ is monoidal with the cartesian product as tensor and the terminal set $1 = \{*\}$ as unit (for simplicity, we consider that it is strictly monoidal). The set $1 \uplus L$ can be equipped with a structure of cubical object $(1 \uplus L, \varepsilon^-, \varepsilon^+, \eta)$ in this category, where $\eta : 1 \to 1 \uplus L$ is the canonical injection and $\varepsilon^-, \varepsilon^+ : 1 \uplus L \to 1$ are both the terminal arrow. Since cubical objects in **Set** are in bijection with monoidal cubical sets, this cubical object corresponds to a cubical set $!\,L$ called the *labeling cubical set* on L, and this operation can be extended into a functor $! : \mathbf{Set}' \to \mathbf{CSet}$ (where $\mathbf{Set}'$ denotes the category of pointed sets). A *labeled cubical set* is an object in the slice category $\mathbf{CSet}\!\downarrow\!!$ of cubical sets over !: a labeled cubical set (C, ℓ, L) consists of a cubical set C and a pointed set L (whose distinguished element is often written $*$) together with a morphism $\ell : C \to !\,L$ of cubical sets. A morphism $(\varphi, \lambda) : (C_1, \ell_1, L_1) \to (C_2, \ell_2, L_2)$ consists of a morphism $\varphi : C_1 \to C_2$ between the underlying cubical sets together with a pointed function $\ell : L_1 \to L_2$ such that $\ell_2 \circ \varphi = \lambda \circ \ell_1$.

As explained earlier, the symmetry of **Set** induces a structure of symmetric cubical set on a pointed set L, which induces a symmetric cubical set that we still write $!\,L$. A *labeled symmetric cubical set* on an alphabet L is an object in the slice category $\mathbf{SCSet}\!\downarrow\!!$. We write **LCSet** (resp. **LSCSet**) for the category of (symmetric) labeled cubical sets and **LPCSet** for the category labeled precubical sets (which is defined similarly).

The cubical set $!\,L$ can be described explicitly: n-cells $l \in \,!\,L(n)$ are lists $l = (e_i)_{1 \leqslant i \leqslant n}$, of length n, of labels $e_i \in L$, where $* \in L$ is a special symbol meaning that the letter is not defined. We write $l_1 \cdot l_2$ for the concatenation of two lists l_1 and l_2. The degeneracy maps $\iota_k : !\,L(n) \to !\,L(n+1)$ send an n-cell $(e_i)_{1 \leqslant i \leqslant n}$ to the list obtained from it by inserting $*$ at the k-th position and the face maps $\partial_k^-, \partial_k^+ : !\,L(n+1) \to !\,L(n)$ both send an $(n+1)$-cell $(e_i)_{1 \leqslant i \leqslant n+1}$ to the list obtained by removing the element at the k-th position. The action of a symmetry $\sigma : n \to n$ on $!\,L(n)$ sends a cell $(e_i)_{1 \leqslant i \leqslant n}$ to $(e_{\sigma(i)})_{1 \leqslant i \leqslant n}$. A label l is said to be *linear* when no event e occurs more than once in l.

3.4 Higher dimensional automata

A *pointed cubical set* (C, i) is a cubical set together with a distinguished 0-cell $i \in C(0)$. A *higher dimensional automaton* (or *HDA*) is a pointed labeled symmetric cubical set C, the distinguished element i being called the *initial state*. A morphism of HDA is a morphism between the underlying labeled symmetric cubical sets which preserves the initial state. Given a category $\mathcal{C}$ of cubical sets, we often write $\mathcal{C}'$ for the corresponding category of pointed cubical sets. We write $\mathbf{HDA} = \mathbf{LSCSet}'$ for the category of HDA and $\mathbf{HDA}_n = \mathbf{LSCSet}'_n$ for the category of n-dimensional HDA.

3.5 Cubical transition systems.

In this section, we introduce a general methodology for associating a cubical set to a model for concurrent processes: we introduce here the notion of cubical transition system, which will help us to generate HDA associated to traditional models. Since monoidal functors preserve the unit of monoidal categories, all cubical sets generated by cubical objects in **Set** (i.e. by the functor !) contain only one 0-cell. Cubical sets with multiple 0-cells can be generated by “actions” of the labeling cubical set on the 0-cells, formalized as follows.

Definition 7. *A* cubical transition system (S, i, E, ℓ, L, t)*, or* CTS*, consists of*
- *a set S of* states*,*

- *a state* $i \in S$ *called the* initial state*,*
- *a set* E *of* events*,*
- *a pointed set* L *of* labels *with* $*$ *as distinguished element,*
- *a* labeling function $\ell : E \to L$*,*
- *a* transition function *which is a partial function* $t : S \times !\,E \to S$

such that for every state x *and every* n*-cell* l *of* $!\,E$ *for which* $t(x, l)$ *is defined,*

1. *if* $l = l_1 \cdot l_2$ *for some cells* l_1 *and* l_2 *then* $t(x, l_1)$ *and* $t(t(x, l_1), l_2)$ *are defined and we have* $t(x, l) = t(t(x, l_1), l_2)$*,*
2. $t(x, ())$ *is defined and equal to* x *(where* $()$ *denotes the 0-cell of* $!\,E$*),*
3. *for every index* $0 \leqslant i \leqslant n$*,* $t(x, !\,E(\eta_i)(l))$ *is defined and equal to* $t(x, l)$*,*
4. *for every symmetry* $\sigma : n \to n$*,* $t(x, !\,E(\sigma)(l))$ *is defined and equal to* $t(x, l)$*.*

An n-cell l of $!\,E$ is *enabled* at a position x if $t(x, l)$ is defined. Every such CTS defines an HDA C labeled by L whose n-cells are pairs (x, l) where x is a state and l is an n-cell of $!\,E$ which is enabled at x. The source and target functions are defined by $\partial_i^-(x, l) = (x, \partial_i^-(l))$ and $\partial_i^+(x, l) = t(t(x, e_i), \partial_i^+(l))$ where e_i is the i-th element of l, the degeneracy maps are defined by $\iota_i(x, l) = (x, !\,E(\eta_i)(l))$ and the symmetries by $\sigma(x, l) = (x, !\,E(\sigma)(l))$. The labeling function is $!\,\ell$ and the initial state is i.

A *morphism* $(\sigma, \tau, \lambda) : (S_1, i_1, E_1, \ell_1, L_1, t_1) \to (S_2, i_2, E_2, \ell_2, L_2, t_2)$ between two CTS consists of a function $\sigma : S_1 \to S_2$, a partial function $\tau : E_1 \to E_2$ and a function $\lambda : L_1 \to L_2$, such that $i_2 = \sigma(i_1)$, $\ell_2 \circ \tau = \lambda \circ \ell_1$, and for every state $x \in S_1$ and cell l of $!\,E_1$, $t_2(\sigma(x), !\,\tau(l)) = \sigma \circ t_1(x, l)$. It can be checked that every such morphism induces a morphism $(\kappa, \lambda) : C_1 \to C_2$ between the corresponding HDA C_1 and C_2 defined by $\kappa(x, l) = (\sigma(x), !\,\tau(l))$. We write **CTS** for the category thus defined.

Theorem 1. *The functor* $\mathbf{CTS} \to \mathbf{HDA}$ *defined above is well-defined.*

3.6 Relating variants of cubical sets.

Suppose that we are given monoidal categories $\mathcal{C}$ and $\mathcal{D}$ and a functor $I : \mathcal{C} \to \mathcal{D}$. Every presheaf $C : \mathcal{D}^{\mathrm{op}} \to \mathbf{Set}$ on $\mathcal{C}$ induces by precomposition with I a presheaf $C \circ I^{\mathrm{op}} : \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$ and this operation can be extended into a functor $\hat{I} : \hat{\mathcal{D}} \to \hat{\mathcal{C}}$ from the presheafs on $\mathcal{D}$ to those on $\mathcal{C}$ defined on morphisms $\alpha : C \to D$ by $(\hat{I}(\alpha))_A = \alpha_{I(A)}$. Using Freyd's adjoint functor theorem [17] (which is also some form of Kan extension in presheaf categories), it can be shown that the functor $\hat{I}$ admits a left adjoint. For example, the inclusion $\boxdot \to \square$ induces a forgetful functor $\mathbf{CSet} \to \mathbf{PCSet}$ which admits a left adjoint. We list here a few interesting such cases:

Lemma 1. *The following forgetful functors admit left adjoints:* $\mathbf{PCSet} \to \mathbf{Set}$*,* $\mathbf{CSet} \to \mathbf{PCSet}$*,* $\mathbf{CSet}_n \to \mathbf{PCSet}_n$*,* $\mathbf{PCSet} \to \mathbf{PCSet}_n$*,* $\mathbf{SCSet} \to \mathbf{CSet}$*, etc. (in particular similar adjunctions hold in the labeled cases).*

These adjoints are very interesting because they allow us to compute for example the free cubical set on a precubical set and so on, and will be used in the following. In particular, the forgetful functor $\mathbf{CSet} \to \mathbf{Set}$ which sends a cubical set C to the set $C(1)$ admits a left adjoint, which can be shown to be the functor ! described in Section 3.3. It can be shown that right adjoints also exist for truncation functors:

Lemma 2. *The functor* $\mathbf{CSet} \to \mathbf{CSet}_n$ *induced by the inclusion* $\square_n \to \square$ *admits a right adjoint* $\mathbf{CSet}_n \to \mathbf{CSet}$ *and similar results hold for other variants of cubical sets, in particular the truncation functor* $\mathbf{HDA} \to \mathbf{HDA}_n$ *admits a right adjoint.*

As an illustration, consider the functor $\mathbf{PCSet} \to \mathbf{PCSet}_n$. Given an n-dimensional precubical set C, the left adjoint sends C to the precubical set D whose k-cells are $D(k) = C(k)$ for $k \leqslant n$ and $D(k) = \emptyset$ otherwise. The action of the right adjoint is more subtle: it sends C to the precubical set obtained from C by "filling in" all the k-dimensional cubes, with $k > n$, by a k-cell.

4 Adjunctions

The purpose of this section is to relate traditional models introduced in Section 2 with the geometric models of Section 3 (mainly HDA).

4.1 Transition systems and HDA

In this section, we relate transition systems and HDA. We begin by relating transition systems to the category of 1-dimensional labeled precubical sets by defining two adjoint functors $F : \mathbf{sTS} \to \mathbf{LPCSet}'_1$ and $G : \mathbf{LPCSet}'_1 \to \mathbf{sTS}$.

We begin by defining the functor F as follows. To any transition system $T = (S, i, E, Tran)$, we associate the pointed 1-dimensional precubical set C labeled by $E \uplus \{*\}$ such that $C(0) = S$, $C(1) = Tran$, the morphisms $\partial_0^- : C_1 \to C_0$ and $\partial_0^+ : C_1 \to C_0$ are respectively $\partial_0^-(s, e, s') = s$ and $\partial_0^+(s, e, s') = s'$, the labeling function is defined by $\ell(s, e, s') = e$ and the distinguished element is $i \in C(0)$. Moreover, to any morphism $(\sigma, \tau) : (S_1, i_1, E_1, Tran_1) \to (S_2, i_2, E_2, Tran_2)$ we associate the morphism (κ, λ) between the labeled precubical sets, where κ is induced by the morphism of graphs whose components are σ on 0-cells and τ on 1-cells, the morphism λ between labels being τ.

Conversely, we define the functor G as follows. To a pointed 1-dimensional precubical set C labeled by L, we associate the transition system $(S, i, E, Tran)$ defined by $S = C(0)$, i being the distinguished element of the pointed precubical set, $E = L$ and the transitions being $Tran = \{\ (\partial_0^-(e), \ell(e), \partial_0^+(e)) \mid e \in E\ \}$. Moreover, to any morphism $(\varphi, \lambda) : C \to D$ between labeled precubical sets, we associate the morphism (σ, τ) defined as $\sigma = \varphi_0 : C(0) \to D(1)$ and $\tau = \lambda$.

A labeled (pre)cubical set (C, ℓ) is *strongly labeled* when there exists no pair of distinct k-cells, for some dimension k, whose sources and targets are equal, which have the same label. The functors defined above enable us to relate both models:

Theorem 2. *The functor* $F : \mathbf{sTS} \to \mathbf{LPCSet}'_1$ *is right adjoint to the functor* $G : \mathbf{LPCSet}'_1 \to \mathbf{sTS}$*. Moreover, the comonad* $G \circ F$ *on* $\mathbf{sTS}$ *is the identity and the adjunction restricts to an equivalence of categories between the full subcategory of* $\mathbf{LPCSet}'_1$ *whose objects are pointed strongly labeled precubical sets.*

Now, recall that the category $\mathbf{TS}$ can be defined as the Kleisli category associated to the monad $(-)_*$ on $\mathbf{sTS}$. Similarly, the adjunction between $\mathbf{LPCSet}_1$ and $\mathbf{LCSet}_1$ given in Lemma 2 induces a monad on $\mathbf{PCSet}_1$ which lifts to a monad T on $\mathbf{PCSet}'_1$. Moreover, it can be shown that $F \circ (-)_* = T \circ F$, $(-)_* \circ G = G \circ T$ and the unit and the multiplication of $(-)_*$ are sent by F to the unit and multiplication of T. From this, we deduce that the associated Kleisli categories are in correspondence [18], i.e.

Theorem 3. *The adjunction of Theorem* (2) *lifts to an adjunction between* **TS** *and* $\mathbf{CSet}'_1$*, which induces an equivalence if we restrict* $\mathbf{CSet}'_1$ *to strongly labeled cubical sets.*

This adjunction is also relevant for HDA since $\mathbf{LCSet}'_1 \cong \mathbf{HDA}_1$. The fact that we have to restrict to a subcategory of $\mathbf{LPCSet}_1$ in Theorem 2 can be explained intuitively by remarking that in transition systems there is no distinction between events and labels: in particular, a transition system cannot contain two distinct transitions with the same event between the same source and the same target. In fact this phenomenon does not arise with labeled transition systems and similar functors show that there is directly an equivalence $\mathbf{LTS} \cong \mathbf{LCSet}'_1 \cong \mathbf{HDA}_1$. All the other models considered here similarly have labeled variants which give rise to better adjunctions. For lack of space, we did not present them here since they are less standard.

4.2 Asynchronous automata and HDA

The adjunction given in previous section, can be extended to an adjunction between asynchronous automata and 2-dimensional HDA. An asynchronous automaton $A = (S, i, E, Tran, I)$ is sent by the right adjoint to a 2-dimensional HDA C, whose underlying 1-dimensional HDA is induced by the underlying transition system of A. The 2-cells are $C(2) = C(1) \uplus C(1) \uplus I$. The cells in $C(1)$ correspond to degenerated 2-cells. Given an element (a_1, s, a_2) in I, there exist transitions (s, a_1, s_1), (s, a_2, s_2), (s_1, a_2, r) and (s_2, a_1, r) and these are uniquely defined by second property of Definition 3: we define face maps by $\partial_0^-(a_1, s, a_2) = (s, a_1, s_1)$, $\partial_0^+(a_1, s, a_2) = (s_2, a_1, r)$, $\partial_1^-(a_1, s, a_2) = (s, a_2, s_2)$ and $\partial_1^+(a_1, s, a_2) = (s_1, a_2, r)$ and the labeling function by $\ell(a_1, s, a_2) = (a_1, a_2)$. The requirement that I is symmetric induces the symmetry of the HDA. The left adjoint is defined similarly. We say that an HDA is n-*deterministic* when two n-cells with the same sources with the same label are equal, and *deterministic* when it is n-deterministic for all $n \geqslant 0$.

Theorem 4. *These two operations can be extended to a functor* $\mathbf{ACR} \to \mathcal{H}$ *which is right adjoint to a functor* $\mathcal{H} \to \mathbf{ACR}$*, where* $\mathcal{H}$ *is the full subcategory of* $\mathbf{HDA}_2$ *whose objects are 1-deterministic HDA. The induced comonad on* **ACR** *is the identity and the adjunction induces an equivalence of categories if we restrict further* $\mathcal{H}$ *to the full subcategory of deterministic HDA.*

By Lemma 1, this induces an adjunction between **ACR** and the full subcategory of **HDA** whose objects are 1-deterministic HDA.

4.3 Event structures and HDA

A *configuration* of an event structure $(E, \leqslant, \#)$ is a finite downward closed subset of compatible events in E. An event e is *enabled* at a configuration x if $e \notin x$ and $x \uplus \{e\}$ is a configuration. An event structure thus induces a CTS (S, i, E, ℓ, L, t) whose states S are the configurations with the empty configuration $\emptyset$ as initial state, labels are $L = E \uplus \{*\}$ with the canonical injection $\ell : E \to E \uplus \{*\}$ as labeling function, transition function t is defined on pairs $(x, e) \in S \times E$ such that e is enabled at x by $t(x, e) = x \uplus \{e\}$ and then extended as a morphism $t : S \times !E \to S$ which is defined only on linear cells of $!E$ (in which an event does not occur more than once). Moreover, every morphism $f : (E, \leqslant, \#) \to (E', \leqslant', \#')$

of event structures induces a morphism $(\sigma, \tau, \lambda) : (S, E, \ell, L, t) \to (S', E', \ell', L', t')$ between the corresponding CTS such that $\sigma : S \to S'$ is the map sending a configuration to the set of images of elements of S by f on which f is defined, $\tau = f$, λ is f. We write $F : \mathbf{ES} \to \mathbf{HDA}$ for the functor thus defined.

Conversely, to every linearly labeled HDA C with i as initial state, we associate the event structure whose events are the labels of C and

- causal dependency is given by $e \leqslant e'$ if for every cell x of C, whose 0-source is i, labeled by a sequence of events l containing e', e occurs before e' in l,
- incompatibility is given by $e \# e'$ if there exists no cell x of C, whose 0-source is i, labeled by a sequence of events l containing both e and e'.

Moreover, to every morphism $(\kappa, \lambda) : C \to D$ of HDA, we associate the morphism f of event structures which to an event e associate $\lambda(e)$ if $\lambda(e) \neq *$ and is not defined otherwise.

Theorem 5. *The functor* $\mathbf{ES} \to \mathcal{H}$ *defined above is right adjoint to the functor* $\mathcal{H} \to \mathbf{ES}$ *where* $\mathcal{H}$ *is the full subcategory of* $\mathbf{HDA}$ *whose objects are linearly labeled HDA and the comonad induced on* $\mathbf{ES}$ *is the identity.*

The adjunction can be boiled down to an equivalence of categories if we restrict further $\mathcal{H}$ to HDA satisfying suitable axioms, in particular the well-known "cube axioms" for asynchronous graphs [25]. As an illustration, the HDA generated by the asynchronous automaton on the left of (4), with $a\, I_x\, c$, $b\, I_x\, c$ and $a\, I_{x_3}\, b$, induces the event structure $(\{a, b, c\}, \leqslant, \#)$, where $\leqslant$ and $\#$ are both the empty relation, which in turn induces the HDA generated by the asynchronous automaton on the right of (4) (where all squares correspond to independence relations).

$$
\begin{tikzcd}
y_1 \arrow[rr, "b"] & & y & \\
& x_3 \arrow[ul, "a"'] \arrow[rr, "b"'] & & y_2 \arrow[ul, "a"'] \\
x_1 \arrow[uu, "c"] & & & \\
& x \arrow[ul, "a"] \arrow[uu, "c"'] \arrow[rr, "b"'] & & x_2 \arrow[uu, "c"']
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
y_1 \arrow[rr, "b"] & & y & \\
& x_3 \arrow[ul, "a"'] \arrow[rr] & & y_2 \arrow[ul, "a"'] \\
x_1 \arrow[uu, "c"] \arrow[rr] & & y_3 \arrow[uu] & \\
& x \arrow[ul, "a"] \arrow[uu] \arrow[rr, "b"'] & & x_2 \arrow[uu, "c"'] \arrow[ul]
\end{tikzcd}
\tag{4}
$$

4.4 Petri nets and HDA

We extend here the adjunction of Winskel and Nielsen [32,5] between 1-bounded Petri Nets and asynchronous transition systems to an adjunction between general Petri Nets and HDA.

From Petri nets to HDA. Suppose given a net $N = (P, M_0, E, \mathrm{pre}, \mathrm{post})$. The pre and post operations can be extended to the cells of $!\,E$ by ${}^\bullet() = {}^\bullet(*) = 0$, ${}^\bullet(e \cdot f) = {}^\bullet e + {}^\bullet f$, $()^\bullet = (*)^\bullet = 0$ and $(e \cdot f)^\bullet = e^\bullet + f^\bullet$. This enables us to see elements of $!\,E$ as generalized events. We also generalize the notion of transition and given two markings M_1 and M_2 and an event $l \in\, !\,E$, we say that there is a transition $M_1 \xrightarrow{l} M_2$ whenever there exists a marking M such that $M_1 = M + {}^\bullet l$ and $M_2 = M + l^\bullet$. In this case, the event l is said to be *enabled* at the marking M_1. The marking M_2 is sometimes denoted M_1/l. A marking M is *reachable* if there exists a transition l such that $M = M_0/l$ where M_0 is the initial marking of N.

Remark 1. As in [32], we have chosen to define morphisms in the opposite direction on places. With the adjunction with HDA in mind, this can be explained as follows.

Morphisms of Petri nets should, just as morphisms of HDA, preserve independence of events: if two events e and e' of a net N are independent and $(\varphi, \psi) : N \to N'$ is a morphism of nets, then their images $\psi(e)$ and $\psi(e')$ should also be independent. By contraposition, this means that if both events $\psi(e)$ and $\psi(e')$ depend on a common place p, then the events e and e' should depend on a corresponding common place $\psi^{-1}(p)$.

Every Petri net N induces a CTS (S, i, E, ℓ, L, t) whose states S are the reachable markings of the net, with the initial marking M_0 as initial state, events E are the events of the net, set of labels is $L = E \uplus \{*\}$ with the canonical injection $L \to L \uplus \{*\}$ as labeling function, transition function $t(M, l)$ is defined if and only if l is enabled at M and in this case $t(M, l) = M/l$. It is routine to verify that this actually defines a CTS and thus an HDA. Moreover, any morphism $(\varphi, \psi) : N \to N'$ between Petri nets induces a morphism (σ, τ, λ) between the corresponding CTS defined by $\tau = \psi$, $\sigma(M) = M \circ \varphi$ for any reachable marking M of N, λ is ψ seen as a total function between pointed sets. We write $\mathrm{hda} : \mathbf{PNet} \to \mathbf{HDA}$ for the functor thus defined.

From HDA to Petri nets. We first introduce the notion of region of an HDA, which should be thought as a way of associating a number of tokens to each 0-cell of the HDA and a pre- and postcondition to every transition of the HDA, in a coherent way. A pre-region R of a set L is a sequence $(R_i)_{i \in \mathbb{N}}$ of functions $R_i : {!}\,L(i) \to \mathbb{N} \times \mathbb{N}$ such that

- for every $e \in {!}\,L(i)$ and $f \in {!}\,L(j)$, $R_{i+j}(e \cdot f) = R_i(e) + R_j(f)$,
- for every $e \in {!}\,L(0)$, $R_0(e) = (0, 0)$.

We often omit the index i since it is determined by the dimension of the cell in argument and respectively write $R'(l)$ and $R''(l)$ for the first and second components of $R(l)$, where l is a cell of ${!}\,L$. Given a cubical set C labeled by L, we implicitly extend a pre-region R to the cells of C by precomposition with the labeling arrow $\ell : C \to {!}\,L$. A *region* of a cubical set labeled by L consists of a pre-region R together with a function $S : C(0) \to \mathbb{N}$ such that for every i-cell $y \in C(i)$ whose 0-source is x and 0-target is x', there exists an integer n such that $(S(x), S(x')) = (n + R'(y), n + R''(y))$.

To every HDA C, we associate a Petri Net $\mathrm{pn}(C)$ whose places are the regions of C, events are the labels of C without the distinguished element, for every event e and place (R, S), ${}^{\bullet}e(R, S) = R'(e)$, $e^{\bullet}(R, S) = R''(e)$, the initial marking M_0 being defined by $M_0(R, S) = S(x_0)$, where x_0 is the initial state of C. Suppose that $(\kappa, \lambda) : C \to D$ is a morphism of HDA. We define a morphism $\mathrm{pn}(\kappa, \lambda) = (\varphi, \psi) : \mathrm{pn}(C) \to \mathrm{pn}(D)$ as follows: φ maps every region (R, S) of D to the region $\varphi(R, S) = (R \circ \kappa, S \circ \kappa_0)$, and ψ is λ seen as a partial function. This thus extends $\mathrm{pn} : \mathbf{HDA} \to \mathbf{PNet}$ into a functor.

The adjunction Suppose that $N = (P, M_0, E, \mathrm{pre}, \mathrm{post})$ is a net and C an HDA labeled by E'. We want to exhibit a bijection between morphisms $\mathrm{pn}(C) \to N$ in $\mathbf{PNet}$ and morphisms $C \to \mathrm{hda}(N)$ in $\mathbf{HDA}$.

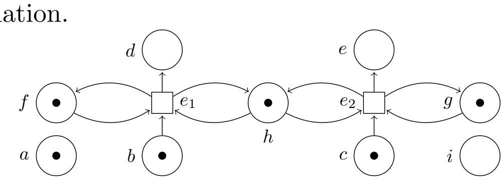
To any morphism $(\varphi, \psi) : \mathrm{pn}(C) \to N$ of nets, we associate a morphism $(\kappa, \lambda) : C \to \mathrm{hda}(N)$ of HDA such that for every 0-cell x of C, $\kappa(x)$ is the marking of N defined on every place p by $\kappa(x)(p) = S_{\varphi(p)}(x)$ (remember that $\varphi(p)$ is a region), and for every n-cell y of C (with $n > 0$), labeled by $l \in {!}\,E$, whose 0-source is x, as the transition $(\kappa(x), {!}\,\psi(l))$ and the morphism λ is ψ.

Conversely, to any morphism $(\kappa, \lambda) : C \to \text{hda}(N)$ of HDA, we associate a morphism $(\varphi, \psi) : \text{pn}(C) \to N$ of nets such that for every place p of N, $\varphi(p)$ is the region of C defined on 0-cells x by $S_{\varphi(p)}(x) = \kappa(x)(p)$ (remember that $\kappa(x)$ is a marking of N) and on labels e by $R_{\varphi(p)}(e) = ((\text{pre} \circ \lambda(e))(p), (\text{post} \circ \lambda(e))(p))$, and on events of C as the morphism λ. It can be shown that these transformations are well defined, are natural in C and N, and are mutually inverse. Therefore,

Theorem 6. *The functor* $\text{hda} : \mathbf{PNet} \to \mathbf{HDA}$ *is right adjoint to the functor* $\text{pn} : \mathbf{HDA} \to \mathbf{PNet}$*.*

If we restrict to 1-bounded nets, which are nets a place can contain either 0 or 1 token, we recover the constructions of [32] for constructing an adjunction between asynchronous transition systems and nets. Since the net associated to an HDA by the functor hda is generally infinite, we will give an example in the case of 1-bounded nets. Consider the asynchronous automaton, depicted on the left of (5), with an empty independence relation.

$$\begin{tikzcd}
& z & \\
y_1 \arrow[ur, "e_2"] & & y_2 \arrow[ul, "e_1"'] \\
& x \arrow[ul, "e_1"] \arrow[ur, "e_2"'] &
\end{tikzcd}$$



(5)

The associated 1-bounded Petri net is shown on the right. In this automaton the place d corresponds to the region (R, S) such that $R(e_1) = (1, 0)$, $R(e_2) = (0, 0)$, $S(x) = S(y_2) = 1$ and $S(y_1) = S(z) = 0$. Now, if we consider the same automaton with $e_1 \, I_x \, e_2$, we obtain the same Petri net with the place h removed.

5 Conclusion and Future work

In this paper, we have made completely formal the relation between HDA and various classical models of concurrent computations: transition systems, asynchronous automata, event structures and Petri nets. This is not only interesting for comparison purposes, between different semantics of parallel languages, but also, for practical reasons, yet to be further studied.

Stubborn sets [28], sleep sets and persistent sets [10] are methods used for diminishing the complexity of model-checking using transition systems. They are based on semantic observations using Petri nets in the first case and Mazurkiewicz trace theory in the other one. We believe that these are special forms of "homotopy retracts" when cast (using the adjunctions we have hinted) in the category of higher-dimensional transition systems. We hope to make this statement more formal through these adjunctions, and use this to design new state-space reduction methods.

Last but not least, in [16] is defined an abstract notion of bisimulation. Given a model for concurrency, i.e. a category of models $\mathbf{M}$ and a "path category" (a subcategory of $\mathbf{M}$ which somehow represents what should be thought of as being paths in the models), then we can define two elements of $\mathbf{M}$ to be bisimilar if there exists a span of special morphisms linking them. These special morphisms have a path-lifting property that, we believe, would be in higher-dimensional transition systems a (geometric) fibration property. We thus hope that homotopy invariants could be useful for the study of a variety of bisimulation equivalences. Some work has been done in that direction in [24,33] (and in some sense also in [13]).

References

1. A. Arnold. *Systèmes de transitions finis et sémantique des processus communicants.* Masson, 1992.
2. M. A. Bednarczyk. *Categories of asynchronous systems.* PhD thesis, 1988.
3. R. Brown and P. J. Higgins. On the algebra of cubes. *JPAA*, (21):233–260, 1981.
4. A. Burroni. Higher-dimensional word problems with applications to equational logic. *Theoretical Computer Science*, 115(1):43–62, 1993.
5. M. Droste and RM Shortt. Petri nets and automata with concurrency relations—an adjunction. In *Sem. of Prog. Lang. and Model Theory*, pages 69–87, 1993.
6. L. Fajstrup. Loops, ditopology, and deadlocks. *Math. Struct. Comput. Sci.*, 2000.
7. L. Fajstrup, E. Goubault, and M. Raußen. Detecting deadlocks in concurrent systems. *CONCUR'98*, pages 332–347.
8. L. Fajstrup, M. Raußen, and E. Goubault. Algebraic topology and concurrency. *Theoretical Computer Science*, 357(1-3):241–278, 2006.
9. L. Fajstrup and S. Sokolowski. Infinitely running processes with loops from a geometric view-point. *ENTCS, Proceedings of GETCO'00*, 2000.
10. P. Godefroid and P. Wolper. Using partial orders for the efficient verification of deadlock freedom and safety properties. volume 575, pages 417–428. LNCS, 1991.
11. E. Goubault. *The Geometry of Concurrency.* PhD thesis, 1995.
12. E. Goubault. Cubical sets are generalized transition systems. Technical report, pre-proceedings of CMCIM'02, 2001.
13. E. Goubault and T. P. Jensen. Homology of higher-dimensional automata. In *Proc. of CONCUR'92*, Stonybrook, New York, August 1992. Springer-Verlag.
14. M. Grandis. *Directed Algebraic Topology; models of non-reversible worlds.* CUP, 2009.
15. M. Grandis and L. Mauri. Cubical sets and their site. *TAC*, 11(8):185–211, 2003.
16. A. Joyal, M. Nielsen, and Winskel G. Bisimulation and open maps. In *LICS*, 1993.
17. S. MacLane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics.* Springer Verlag, 1971.
18. P. Mulry. Lifting theorems for Kleisli categories. In *MFPS*, pages 304–319, 1994.
19. M. Nielsen, G. D. Plotkin, and G. Winskel. Petri nets, event structures and domains. In *Semantics of Concurrent Computation*, pages 266–284, 1979.
20. C. Petri. Communication with automata, 1966.
21. V. Pratt. Modeling concurrency with geometry. In *Proc. of the 18th ACM Symposium on Principles of Programming Languages.* ACM Press, 1991.
22. M. Raussen. On the classification of dipaths in geometric models for concurrency. *Mathematical Structures in Computer Science*, August 2000.
23. Sankaranarayanan S., H. Sipma, and Z. Manna. Petri net analysis using invariant generation. In *In Verification: Theory and Practice*, volume 2772 of *LNCS*, pages 682–701, 2003.
24. V. Sassone and G. L. Cattani. Higher-dimensional transition systems. In *LICS*, 1996.
25. V. Sassone, M. Nielsen, and G. Winskel. Relationships between models of concurrency. In *Proceedings of the Rex'93 school and symposium*, 1994.
26. J.P. Serre. *Homologie Singulière des Espaces Fibrés. Applications.* PhD thesis, École Normale Supérieure, 1951.
27. M.W. Shields. Concurrent machines. *Computer Journal*, 28, 1985.
28. A. Valmari. A stubborn attack on state explosion. In *Proc. of CAV'90.* Springer Verlag, LNCS, 1990.
29. R. van Glabbeek. Bisimulation semantics for higher dimensional automata. Technical report, Stanford University, 1991.
30. R.J. Van Glabbeek. Petri nets, configuration structures and higher dimensional automata. *Lecture notes in computer science*, pages 21–27, 1999.
31. G. Winskel. Event structures. In *Advances in Petri Nets*, pages 325–392, 1986.
32. G. Winskel and M. Nielsen. Models for concurrency. In *Handbook of Logic in Computer Science*, volume 3, pages 1–148. Oxford University Press, 1995.
33. K. Worytkiewicz. Paths and simulations. *ENTCS*, 69:346–361, 2003.