



Trees in Partial Higher Dimensional Automata

Jérémy Dubut^{1,2}(✉)

¹ National Institute of Informatics, Tokyo, Japan
dubut@nii.ac.jp

² Japanese-French Laboratory for Informatics, Tokyo, Japan

Abstract. In this paper, we give a new definition of partial Higher Dimension Automata using lax functors. This definition is simpler and more natural from a categorical point of view, but also matches more clearly the intuition that pHDA are Higher Dimensional Automata with some missing faces. We then focus on trees. Originally, for example in transition systems, trees are defined as those systems that have a unique path property. To understand what kind of unique property is needed in pHDA, we start by looking at trees as colimits of paths. This definition tells us that trees are exactly the pHDA with the unique path property modulo a notion of homotopy, and without any shortcuts. This property allows us to prove two interesting characterisations of trees: trees are exactly those pHDA that are an unfolding of another pHDA; and trees are exactly the cofibrant objects, much as in the language of Quillen's model structure. In particular, this last characterisation gives the premisses of a new understanding of concurrency theory using homotopy theory.

Keywords: Higher Dimensional Automata · Trees · Homotopy theories

1 Introduction

Higher Dimensional Automata (HDA, for short), introduced by Pratt in [23], are a geometric model of true concurrency. Geometric, because they are defined very similarly to simplicial sets, and can be interpreted as glueings of geometric objects, here hypercubes of any dimension. Similarly to other models of concurrency much as event structures [21], asynchronous systems [1, 25], or transition systems with independence [22], they model true concurrency, in the sense that they distinguish interleaving behaviours from simultaneous behaviours. In [12], van Glabbeek proved that they form the most powerful models of a hierarchy of concurrent models. In [6], Fahrenberg described a notion of bisimilarity of HDA using the general framework of open maps from [17]. If this work is very natural,

The author was supported by ERATO HASUO Metamathematics for Systems Design 27 Project (No. JPMJER1603), JST.

© The Author(s) 2019

M. Bojańczyk and A. Simpson (Eds.): FOSSACS 2019, LNCS 11425, pp. 224–241, 2019.

https://doi.org/10.1007/978-3-030-17127-8_13

it is confronted with a design problem: paths (or executions) cannot be nicely encoded as HDA. Indeed, in a HDA, it is impossible to model the fact that two actions *must* be executed at the same time, or that two actions are executed at the same time but one *must* start before the other. From a geometric point of view, those impossibilities are expressed by the fact that we deal with closed cubes, that is, cubes that must contain all of their faces. Motivated by those examples, Fahrenberg, in [7], extended HDA to partial HDA, intuitively, HDA with cubes with some missing faces. If the intuition is clear, the formalisation is still complicated to achieve: the definition from [7] misses the point that faces can be not uniquely defined. This comes from the fact that Fahrenberg wanted to stick to the ‘local’ definition of precubical sets, that is, that cubes must satisfy some local conditions about faces. As we will show, those local equations are not enough in the partial case. Another missed point is the notion of morphisms of partial HDA: as defined in [7], the natural property that morphisms map executions to executions is not satisfied. In Sect. 2, we address those issues by giving a new definition of partial HDA in terms of lax functors. This definition, similar to the presheaf theoretic definition of HDA, avoid the issues discussed above by considering global inclusions, instead of local equations. This illustrates more clearly the intuition of partial HDA being HDA with missing faces: we coherently replace sets and total functions by sets and partial functions. From this similarity with the original definition of HDA, we can prove that it is possible to complete a partial HDA to turn it into a HDA, by adding the missing faces, and from this completion, it is possible to define a geometric realisation of pHDA (which was impossible with Fahrenberg’s definition).

The geometry of Higher Dimensional Automata, and more generally, of true concurrency, has been studied since Goubault’s PhD thesis [13]. Since then, numerous pieces of work relating algebraic topology and true concurrency have been achieved (for example, see the textbooks [9, 14]). In particular, some attempts of defining nice homotopy theories for true concurrency (or directed topology), through the language of model structures of Quillen [24], have been made by Gaucher [10], and the author [3]. In the second part of this paper (Sects. 3, 4 and 5), we consider another point of view of this relationship between HDA and model structures. The goal is not to understand the true concurrency of HDA, that is, understanding the homotopy theory of HDA as an abstract homotopy theory, but to understand the concurrency theory of HDA. By this we mean to understand how paths (or executions) and extensions of paths can be understood using (co)fibrations (in Quillen’s sense). Also, the goal is not to construct a model structure, as Quillen’s axioms would fail, but to give intuitions and some preliminary formal statements toward the understanding of concurrency using homotopy theory. Using this point of view, many constructions in concurrency can be understood using the language of model structures:

- Open maps from [17] can be understood as trivial fibrations, namely weak equivalences (here, bisimulations) that have the right lifting properties with respect to some morphisms.

- Those morphisms are precisely extensions of executions, which means that they can be seen as cofibration generators (in the language of cofibrantly generated model structures [15]).
- Cofibrations are then morphisms that have the left lifting property with respect to open maps. In particular, this allows us to define cofibrant objects as those objects whose unique morphism from the initial object is a cofibration. In a way, cofibrant objects are those objects that are constructed by just using extensions of paths, and should correspond to trees.
- The cofibrant replacement is then given by canonically constructing a cofibrant object, which is weakly equivalent (here, bisimilar) to a given object. That should correspond to the unfolding.

The main ingredient is to understand what trees are in this context. In the case of transition systems for semantics of CCS [19], synchronisation trees are those systems with exactly one path from the initial state to any state. Those trees are then much simpler to reason on, but they are still powerful enough to capture any bisimulation type: by unfolding, it is possible to canonically construct a tree from a system. The goal of Sects. 3 and 4 will be to understand how to generalise this to pHDA. In this context, it is not clear what kind of unique path property should be considered as, in general, in truly concurrent systems, we have to deal with homotopies, namely, equivalences of paths modulo permutation of independent actions. Following [4], we will first consider trees as colimits of paths. This will guide us to determine what kind of unique path property is needed: a tree is a pHDA with exactly one class of paths modulo a notion of homotopy, from the initial state to any state, and without any shortcuts. This will be proved by defining a suitable notion of unfolding of pHDA. Finally, in Sect. 5, we prove that those trees coincide exactly with the cofibrant objects, illustrating the first steps of this new understanding of concurrency, using homotopy theory.

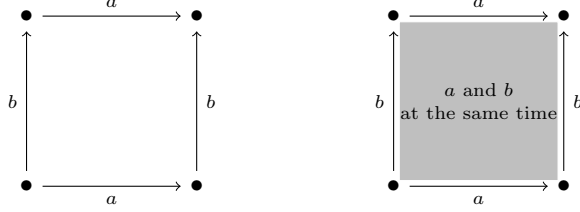
2 Fixing the Definition of pHDA

In this Section, we review the definitions of HDA (Sect. 2.1), the first one using face maps, and the second one using presheaves. In Sect. 2.2, we describe the definition of partial HDA from [7] and explain why it does not give us what we are expecting. We tackle those issues by introducing a new definition in Sect. 2.3, extending the presheaf theoretic definition, using lax functors instead of strict functors. Finally, in Sect. 2.4, we prove that HDA form a reflective subcategory of partial HDA, by constructing a completion of a partial HDA.

2.1 Higher Dimensional Automata

Higher Dimensional Automata are an extension of transition systems: they are labeled graphs, except that, in addition to vertices and edges, the graph structure also has higher dimensional data, expressing the fact that several actions can be made at the same time. Those additional data are intuitively cubes filling up interleaving: if a and b can be made at the same time, instead of having an

empty square as on the left figure, with $a.b$ and $b.a$ as only behaviours, we have a full square as on the right figure, with any possible behaviours in-between. This requires to extend the notion of graph to add those higher dimensional cubical data: that is the notion of *precubical sets*.



Concrete Definition of Precubical Sets. A **precubical set** X is a collection of sets $(X_n)_{n \in \mathbb{N}}$ together with a collection of functions $(\partial_{i,n}^\alpha : X_n \rightarrow X_{n-1})_{n > 0, 1 \leq i \leq n, \alpha \in \{0,1\}}$ satisfying the local equations $\partial_{i,n}^\alpha \circ \partial_{j,n+1}^\beta = \partial_{j,n}^\beta \circ \partial_{i+1,n+1}^\alpha$ for every $\alpha, \beta \in \{0,1\}$, $n > 0$ and $1 \leq j \leq i \leq n$. A **morphism of precubical sets** from X to Y is a collection of functions $(f_n : X_n \rightarrow Y_n)_{n \in \mathbb{N}}$ satisfying the equations $f_n \circ \partial_{i,n}^\alpha = \partial_{i,n}^\alpha \circ f_{n+1}$ for every $n \in \mathbb{N}$, $1 \leq i \leq n$ and $\alpha \in \{0,1\}$. The elements of X_0 are called **points**, X_1 **segments**, X_2 **squares**, X_n **n -cubes**. In the following, we will call **past** (resp. **future**) **i -face maps** the $\partial_{i,n}^0$ (resp. $\partial_{i,n}^1$). We denote this category of precubical sets by **pCub**.

Precubical Sets as Presheaves. Equivalently, **pCub** is the category of presheaves over the cubical category \square . \square is the subcategory of **Set** whose objects are the sets $\{0,1\}^n$ for $n \in \mathbb{N}$ and whose morphisms are generated by the so-called **coface maps**:

$$d_{i,n}^\alpha : \{0,1\}^{n-1} \rightarrow \{0,1\}^n \quad (\beta_1, \dots, \beta_{n-1}) \mapsto (\beta_1, \dots, \beta_{i-1}, \alpha, \beta_i, \dots, \beta_{n-1})$$

A precubical set is a functor $X : \square^{op} \rightarrow \mathbf{Set}$, that is, a presheaf over \square , and a morphism of precubical sets is a natural transformation.

Higher Dimensional Automata [11]. From now on, fix a set L , called the **alphabet**. We can form a precubical set also noted L such that $L_n = L^n$ and the i -face maps are given by $\delta_i^\alpha(a_1 \dots a_n) = a_1 \dots a_{i-1}.a_{i+1} \dots a_n$. We can also form the following precubical set $*$ such that $*_0 = \{*\}$ and $*_n = \emptyset$ for $n > 0$. A **HDA** X on L is a bialgebra $* \rightarrow X \rightarrow L$ in **pCub**. In other words, a HDA X is a precubical set, also noted X , together with a specified point, the **initial state**, $i \in X_0$ and a **labelling function** $\lambda : X_1 \rightarrow L$ satisfying the equations $\lambda \circ \partial_{i,2}^0 = \lambda \circ \partial_{i,2}^1$ for $i \in \{1,2\}$ (see previous figure, right). A **morphism of HDA** from X to Y is a morphism f of precubical sets from X to Y such that $f_0(i_X) = i_Y$ and $\lambda_X = \lambda_Y \circ f_1$. HDA on L and morphisms of HDA form a category that we denote by **HDA_L**. This category can also be defined as the double slice category $*/\mathbf{pCub}/L$. Remark that we are only concerned with labelling-preserving morphisms, not general morphisms as described in [5].

2.2 Original Definition of Partial Higher Dimensional Automata

Originally [7], partial HDA are defined similarly to the concrete definition of HDA, except that the face maps can be partial functions and the local equations hold only when *both* sides are well defined. There are two reasons why it fails to give the good intuition:

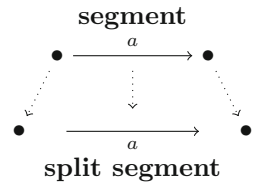
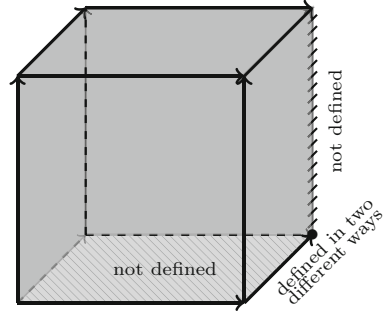
– first the ‘local’ equations are not enough in the partial case. Imagine that we want to model a full cube c without its lower face, that is, $\partial_{3,3}^0$ is not defined on c , and such that $\partial_{1,2}^1$ is undefined on $\partial_{1,3}^1(c)$ and $\partial_{2,3}^1(c)$, that is, we remove an edge. We cannot prove using the local equations that $\partial_1^1 \circ \partial_2^0 \circ \partial_1^1(c) = \partial_1^1 \circ \partial_2^0 \circ \partial_2^1(c)$, that is, that the vertices of the cube are uniquely defined. Indeed, to prove this equality using the local equations, you can only permute two consecutive ∂ . From $\partial_1^1 \circ \partial_2^0 \circ \partial_1^1(c)$, you can:

- either permute the first two and you obtain $\partial_1^1 \circ \partial_1^1 \circ \partial_3^0(c)$,
- or permute the last two and you obtain $\partial_1^1 \circ \partial_1^1 \circ \partial_1^1(c)$.

and both faces are not defined. On the other hand, those two should be equal because the comaps $d_1^1 \circ d_2^0 \circ d_1^1$ and $d_2^1 \circ d_2^0 \circ d_1^1$ are equal in \square , and $\partial_1^1 \circ \partial_2^0 \circ \partial_1^1$ and $\partial_1^1 \circ \partial_2^0 \circ \partial_2^1$ are both defined on c .

– secondly, the notion of morphism is not good (or at least, ambiguous). The equations $f_n \circ \partial_{i,n,X}^\alpha = \partial_{i,n,Y}^\alpha \circ f_{n+1}$ hold in [7] only when *both* face maps are defined, which authorises many morphisms. For example, consider the segment I , and the ‘split’ segment I' which is defined as I , except that no face maps are defined (geometrically, this corresponds to two points and an open segment). The

identity map from I to I' is a morphism of partial precubical sets in the sense of [7], which is unexpected. A bad consequence of that is that the notion of paths in a partial HDA does not correspond to morphisms from some particular partial HDA, and paths are not preserved by morphisms, as we will see later.



2.3 Partial Higher Dimensional Automata as Lax Functors

The idea is to generalise the ‘presheaf’ definition of precubical sets. The problem is to deal with partial functions and when two of them should coincide. Let \mathbf{pSet} be the category of sets and partial functions. A partial function $f : X \rightarrow Y$ can be either seen as a pair (A, f) of a subset $A \subseteq X$ and a total function $f : A \rightarrow Y$, or as a functional relation $f \subseteq X \times Y$, that is, a relation such that for every $x \in X$, there is at most one $y \in Y$ with $(x, y) \in f$. We will freely use both views in the following. For two partial maps $f, g : X \rightarrow Y$, we denote by $f \equiv g$ if and only if for every $x \in X$ such that $f(x)$ and $g(x)$ are defined, then

$f(x) = g(x)$. Note that this is not equality, but equality on the intersection of the domains. We also write $f \subseteq g$ if and only if f is include in g as a relation, that is, if and only if, for every $x \in X$ such that $f(x)$ is defined, then $g(x)$ is defined and $f(x) = g(x)$. By a **lax functor** $F : \mathcal{C} \rightarrow \mathbf{pSet}$, we mean the following data [20]:

- for every object c of \mathcal{C} , a set Fc ,
- for every morphism $i : c \longrightarrow c'$, a partial function $Fi : Fc \longrightarrow Fc'$

satisfying that $Fid_c = id_{Fc}$ and $Fj \circ Fi \subseteq F(j \circ i)$.

The point is that partial precubical sets as defined in [7] do not satisfy the second condition, while they should. In addition, this definition will authorise a square to have vertices, that is, that some $\partial\partial$ are defined, while having no edge, that is, no ∂ defined. This may be useful to define paths as discrete traces in [8] (that we will call *shortcuts* later), that is, paths that can go directly from a point to a square for example. Observe also that if $j \circ i = j' \circ i'$ then $Fj \circ Fi \equiv Fj' \circ Fi'$, which gives us the local equations from [7]. A **partial precubical set** X is then a lax functor $F : \square^{op} \rightarrow \mathbf{pSet}$. It becomes harder to describe explicitly what a partial precubical set is, since we cannot restrict to the ∂_i^α anymore. It is a collection of sets $(X_n)_{n \in \mathbb{N}}$ together with a collection of *partial* functions $(\partial_{i_1 < \dots < i_k}^{\alpha_1, \dots, \alpha_k} : X_{n+k} \longrightarrow X_n)$ satisfying the inclusions $\partial_{j_1 < \dots < j_m}^{\beta_1, \dots, \beta_m} \circ \partial_{i_1 < \dots < i_n}^{\alpha_1, \dots, \alpha_n} \subseteq \partial_{k_1 < \dots < k_{n+m}}^{\gamma_1, \dots, \gamma_{n+m}}$ where the k_s and γ_s are defined as follows. $(k_1 < \dots < k_{n+m}; \gamma_1, \dots, \gamma_{n+m}) = (i_1 < \dots < i_n; \alpha_1, \dots, \alpha_n) \star (j_1 < \dots < j_m; \beta_1, \dots, \beta_m)$ where \star is defined by induction on $n + m$:

- if $n = 0$, $\epsilon \star (j_1 < \dots < j_m; \beta_1, \dots, \beta_m) = (j_1 < \dots < j_m; \beta_1, \dots, \beta_m)$,
- if $m = 0$, $(i_1 < \dots < i_n; \alpha_1, \dots, \alpha_n) \star \epsilon = (i_1 < \dots < i_n; \alpha_1, \dots, \alpha_n)$,
- if $i_1 \leq j_1$, $(i_1 < \dots < i_n; \alpha_1, \dots, \alpha_n) \star (j_1 < \dots < j_m; \beta_1, \dots, \beta_m) = (i_1; \alpha_1).((i_2 < \dots < i_n; \alpha_2, \dots, \alpha_n) \star (j_1 + 1 < \dots < j_m + 1; \beta_1, \dots, \beta_m))$,
- if $i_1 > j_1$, $(i_1 < \dots < i_n; \alpha_1, \dots, \alpha_n) \star (j_1 < \dots < j_m; \beta_1, \dots, \beta_m) = (j_1; \beta_1).((i_1 < \dots < i_n; \alpha_1, \dots, \alpha_n) \star (j_2 < \dots < j_m; \beta_2, \dots, \beta_m))$.

A **function-valued op-lax transformation** [20] from $F : \mathcal{C} \rightarrow \mathbf{pSet}$ to $G : \mathcal{C} \rightarrow \mathbf{pSet}$ is a collection $(f_c)_{c \in Ob(\mathcal{C})}$ of *total* functions such that for every $i : c \longrightarrow c'$, $f_{c'} \circ F(i) \subseteq G(i) \circ f_c$. A **morphism of partial precubical sets** from X to Y is then a function-valued op-lax transformation. In other words, this is a collection of *total* functions $(f_n : X_n \longrightarrow Y_n)_{n \in \mathbb{N}}$ satisfying the equations $f_n \circ \partial_{i_1 < \dots < i_k}^{\alpha_1, \dots, \alpha_k} \subseteq \partial_{i_1 < \dots < i_k}^{\alpha_1, \dots, \alpha_k} \circ f_{n+k}$. Partial precubical sets and morphisms of partial precubical sets form a category that we denote by **ppCub**. **pCub** is a full subcategory of **ppCub**. In particular, the precubical sets $*$ and L are partial precubical sets. A **partial HDA** X on L is a partial precubical set, also noted X , together with a specified point, the **initial state** $i \in X_0$ and a morphism of **ppCub**, the **labelling functions**, $(\lambda_n : X_n \longrightarrow L^n)_{n \in \mathbb{N}}$. A **morphism of pHDA** from X to Y is a morphism f of partial precubical sets from X to Y such that $f_0(i_X) = i_Y$ and $\lambda_X = \lambda_Y \circ f$. Partial HDA on L and morphisms of partial HDA form a category that we note **pHDA_L**. In other words, this is the double slice category $*/\mathbf{ppCub}/L$.

2.4 Completion of a pHDA

Let us describe how it is possible to construct a HDA from a pHDA X , by ‘completing’ X , that is, by adding the faces that are missing, and by connecting the faces that are not. Let

$$Y_n = \{((i_1 < \dots < i_k; \alpha_1, \dots, \alpha_k), x) \mid x \in X_{n+k} \wedge i_k \leq n+k\}$$

$Y = (Y_n)_{n \in \mathbb{N}}$ is intuitively the collection of all abstract faces of all cubes of X , that is, pairs of a cube and all possible ways to define a face from it. Of course, some of those are the same, since there are several ways to describe a cube as the face of some other cube. Define \sim as the smallest equivalence relation such that:

- if $\partial_{i_1 < \dots < i_k}^{\alpha_1, \dots, \alpha_k}(x)$ is defined, then

$$((i_1 < \dots < i_k; \alpha_1, \dots, \alpha_k), x) \sim (\epsilon, \partial_{i_1 < \dots < i_k}^{\alpha_1, \dots, \alpha_k}(x)).$$

This means that, if a face of a cube exists in X , this face is identified with both abstract faces $(\epsilon, \partial_{i_1 < \dots < i_k}^{\alpha_1, \dots, \alpha_k}(x))$ (i.e., the cube $\partial_{i_1 < \dots < i_k}^{\alpha_1, \dots, \alpha_k}(x)$ itself) and $((i_1 < \dots < i_k; \alpha_1, \dots, \alpha_k), x)$ (i.e., the face of x , which consists of taking the (i_k, α_k) face, then the (i_{k-1}, α_{k-1}) face, and so on).

- if $((i_1 < \dots < i_k; \alpha_1, \dots, \alpha_k), x) \sim ((j_1 < \dots < j_l; \beta_1, \dots, \beta_l), y)$, then $((i_1 < \dots < i_k; \alpha_1, \dots, \alpha_k) \star (i, \alpha), x) \sim ((j_1 < \dots < j_l; \beta_1, \dots, \beta_l) \star (i, \alpha), y)$. This means that if two abstract faces coincide, then taking both their (i, α) face gives two abstract faces that also coincide.

Let $\chi(X)_n = Y_n / \sim$ and we denote by $\ll (i_1 < \dots < i_k; \alpha_1, \dots, \alpha_k), x \gg$ the equivalence class of $((i_1 < \dots < i_k; \alpha_1, \dots, \alpha_k), x)$ modulo \sim . We define the i -face map as $\partial_i^\alpha(\ll (i_1 < \dots < i_k; \alpha_1, \dots, \alpha_k), x \gg) = \ll (i_1 < \dots < i_k; \alpha_1, \dots, \alpha_k) \star (i, \alpha), x \gg$, the initial state as $\ll \epsilon, i \gg$ and the labelling function as $\lambda(\ll (i_1 < \dots < i_k; \alpha_1, \dots, \alpha_k), x \gg) = \delta_{i_1}^{\alpha_1} \circ \dots \circ \delta_{i_k}^{\alpha_k}(\lambda(x))$.

Theorem 1. χ is a well-defined functor and is the left adjoint of τ , the injection of \mathbf{HDA}_L into \mathbf{pHDA}_L . Furthermore, \mathbf{HDA}_L is a reflective subcategory of \mathbf{pHDA}_L .

Now, we can define the **geometric realisation** of a pHDA X as the subspace of the realisation of $\chi(X)$ consisting of points whose carrier is of the form $\ll \epsilon, x \gg$ for some $x \in X$. This really corresponds to the drawings we have been using to depict pHDA until now.

3 Paths in Partial Higher Dimensional Automata

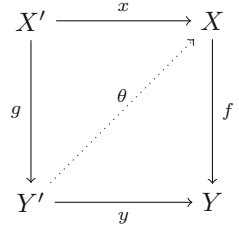
Executions of HDA are defined using the notion of paths. Those paths describe the succession of starting and finishing of actions in a HDA. For example, a HDA can start an action then start another at the same time, and finish the two

actions. This sequence is then not just a sequence of 1-dimensional transitions, since some actions can be made at the same time, but a sequence of hypercubes corresponding to the evolution of the state of the system. We will formalise this idea in Sect. 3.2, and we will see in particular that those paths can be encoded in the category $\mathbf{pHDA}_{\mathbf{L}}$ (while it is not possible in the category $\mathbf{HDA}_{\mathbf{L}}$) as morphisms from particular pHDA, called path shapes. In Sect. 3.1, let us first start by recalling the general framework of [17].

3.1 Path Category, Open Maps, Coverings

In the general framework of [17], we start with a category \mathcal{M} of systems, together with a subcategory \mathcal{P} of execution shapes. For example, keep in mind the case where \mathcal{M} is the category of transition systems and \mathcal{P} is the full subcategory of finite linear systems. One interesting remark about this case is that executions of a given systems are in bijective correspondance with morphisms from a finite linear system to this given system. This means that to reason about behaviours of such systems, it is enough to reason about morphisms and execution shapes.

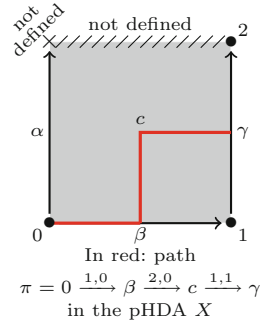
This idea was formalised by describing precisely which morphisms are witnesses for the existence of a bisimulation between systems. This description uses right lifting properties: we say that a morphism $f : X \rightarrow Y$ has the **right lifting property with respect to** $g : X' \rightarrow Y'$ if for every $x : X' \rightarrow X$ and $y : Y' \rightarrow Y$ such that $f \circ x = y \circ g$, there exists $\theta : Y' \rightarrow X$ such that $x = \theta \circ g$ and $f \circ \theta = y$. For example, let us assume that f is a morphism of transition systems and that X' and Y' are finite linear systems. Then x (resp. y) is the same as an execution in X (resp. Y), and $f \circ x = y \circ g$ means that the execution y is an extension of the image of the execution x by f . The right lifting property means that the longer execution y of Y can be lifted to a longer execution θ of X , that is, θ is an extension of x and the image of θ by f is y . This property of lifting longer executions is precisely the property needed on a morphism to make its graph relation a bisimulation. They are also very similar to morphisms of coalgebras [16]. We call **\mathcal{P} -open** (or simply open when \mathcal{P} is clear), a morphism that has the right lifting property with respect to every morphism in \mathcal{P} . From open maps, it is possible to describe similarity and bisimilarity as the existence of a span of morphisms/open maps, and many kinds of bisimilarities can be captured in this way [17]. An open map is said to be a **\mathcal{P} -covering** (or simply covering) if furthermore the lifts in the right lifting properties are unique. Being a covering is a very strong requirement, as they correspond to partial unfolding of a system.



3.2 Encoding Paths in pHDA

In this section, we describe the classical notion of execution of HDA from [12], extended to partial HDA in [7], defined using the notion of path. We then show that those executions can be encoded as an execution shapes subcategory, as in the general framework of [17], proving in particular that paths are in bijective correspondence with a class of morphisms. A **path** π of a HDA X is a sequence $i = x_0 \xrightarrow{j_1, \alpha_1} x_1 \xrightarrow{j_2, \alpha_2} \dots \xrightarrow{j_n, \alpha_n} x_n$ where $x_k \in X$, $j_k > 0$ and $\alpha_k \in \{0, 1\}$ are such that for every k :

- if $\alpha_k = 0$, then $x_{k-1} = \partial_{j_k}^0(x_k)$,
- if $\alpha_k = 1$, then $x_k = \partial_{j_k}^1(x_{k-1})$.



This definition can easily be extended to pHDA, by requiring that the j_k -face maps are defined on x_k or x_{k-1} . A natural property of executions and morphisms is that morphisms map executions to executions. This is the case here (while it is not for [7], e.g., the split segment):

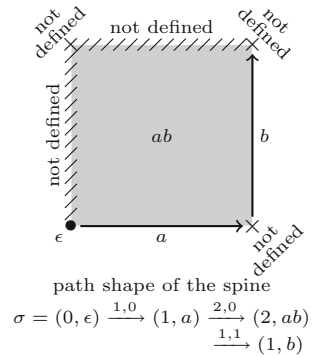
Proposition 1. *If $f : X \longrightarrow Y$ is a map of pHDA and if $\pi = x_0 \xrightarrow{j_1, \alpha_1} x_1 \xrightarrow{j_2, \alpha_2} \dots \xrightarrow{j_n, \alpha_n} x_n$ is a path in X , then $\pi' = f(x_0) \xrightarrow{j_1, \alpha_1} f(x_1) \xrightarrow{j_2, \alpha_2} \dots \xrightarrow{j_n, \alpha_n} f(x_n)$ is a path in Y .*

One advantage of considering pHDA instead of HDA is that paths can be encoded in pHDA, which is not really possible in HDA. It is done as follows. A **spine** σ is a sequence $(0, \epsilon) = (d_0, w_0) \xrightarrow{j_1, \alpha_1} (d_1, w_1) \xrightarrow{j_2, \alpha_2} \dots \xrightarrow{j_n, \alpha_n} (d_n, w_n)$ where $j_k > 0$, $d_k \in \mathbb{N}$, $w_k \in L^{d_k}$ and $\alpha_k \in \{0, 1\}$ are such that:

- if $\alpha_k = 0$, then $d_{k-1} = d_k - 1$, $\delta_{j_k}(w_k) = w_{k-1}$ and $j_k \leq d_k$,
- if $\alpha_k = 1$, then $d_k = d_{k-1} - 1$, $\delta_{j_k}(w_{k-1}) = w_k$ and $j_k \leq d_{k-1}$.

A path π has a underlying spine σ_π by mapping x_k to the pair of its dimension and its label. A spine σ induces a pHDA $B\sigma$ as follows:

- $B\sigma_p = \{k \in \{0, \dots, n\} \mid d_k = p\}$,
- the partial face maps $\partial_{i_1 < \dots < i_n}^{\alpha_1, \dots, \alpha_n}$ are the smallest (as relations ordered by inclusion) partial functions such that:
 - if $\alpha_k = 0$, then $\partial_{j_k}^0(k) = k - 1$,
 - if $\alpha_k = 1$, then $\partial_{j_k}^1(k - 1) = k$,
 - $\partial_{j_1, \dots, j_m}^{\beta_1, \dots, \beta_m} \circ \partial_{i_1 < \dots < i_n}^{\alpha_1, \dots, \alpha_n} \subseteq \partial_{k_1 < \dots < k_{n+m}}^{\gamma_1, \dots, \gamma_{n+m}}$, for $(k_1, \dots, k_{n+m}; \gamma_1, \dots, \gamma_{n+m}) = (i_1, \dots, i_n; \alpha_1, \dots, \alpha_n) \star (j_1, \dots, j_m; \beta_1, \dots, \beta_m)$.



- the initial state is 0,
- the labelling functions λ_n map k to w_k .

By a **path shape**, we mean such a pHDA $B\sigma$. The set **Spine_L** of spines can be partially ordered by prefix. B can then be extended to an embedding from **Spine_L** to **pHDA_L**. We note **PS_L** the image of this embedding, i.e., the full sub-category of path shapes.

Proposition 2. *There is a bijection between paths in a pHDA X and morphisms of pHDA from a path shape to X .*

Again, this is not true with the definition of morphisms from [7] (e.g., the split segment). As an example, the red path π above corresponds to a morphism from the path shape $B\sigma$ to X .

4 Trees and Unfolding in pHDA

In this section, we introduce our notion of trees. Following [4], we consider trees as colimits (or glueings of paths). Section 4.1 is dedicated to proving that those colimits actually exist, by giving an explicit construction of those. From this explicit construction, we will describe the kind of unique path properties that are satisfied by those trees in Sect. 4.2. Starting by showing, that the strict unicity of path fails, we then describe a notion of homotopy, the confluent homotopy, which is weaker than the one from [12], for which every tree has the property that there is exactly one homotopy class of paths from the initial state to any state. We will also see that, because the face maps of trees are defined in a local way, they do not have any shortcuts, that is, paths that ‘skip’ dimensions, for example, going from a point to a square without going through a segment. Finally, in Sect. 4.3, we will prove that those two properties – the unicity of paths modulo confluent homotopy, and the non-existence of shortcuts – completely characterise trees. This proof will use a suitable notion of unfolding of pHDA, showing furthermore that trees form a coreflective subcategory of pHDA.

4.1 Trees, as Colimits of Paths in pHDA

In this section, we give an explicit construction of colimits of diagrams with values in path shapes. Those will be our first definition of trees in pHDA, following [4]. Let $D : \mathcal{C} \longrightarrow \mathbf{PS}_L$ be a small diagram with values in **PS_L**, that is, a functor from \mathcal{C} to **PS_L**. Let us use some notations: for every object u of \mathcal{C} , $Du = B\sigma_u$ with $\sigma_u = (d_0^u, w_0^u) \xrightarrow{j_1^u, \alpha_1^u} (d_1^u, w_1^u) \xrightarrow{j_2^u, \alpha_2^u} \dots \xrightarrow{j_{l_u}^u, \alpha_{l_u}^u} (d_{l_u}^u, w_{l_u}^u)$. The definition of the colimit $\text{col } D$ will be in two steps. The first step consists in putting all the paths Du side-by-side, and in glueing them together, along the morphisms Df , for every morphism f of \mathcal{C} . This is done as follows. Define $(X_n)_{n \in \mathbb{N}}$ to be:

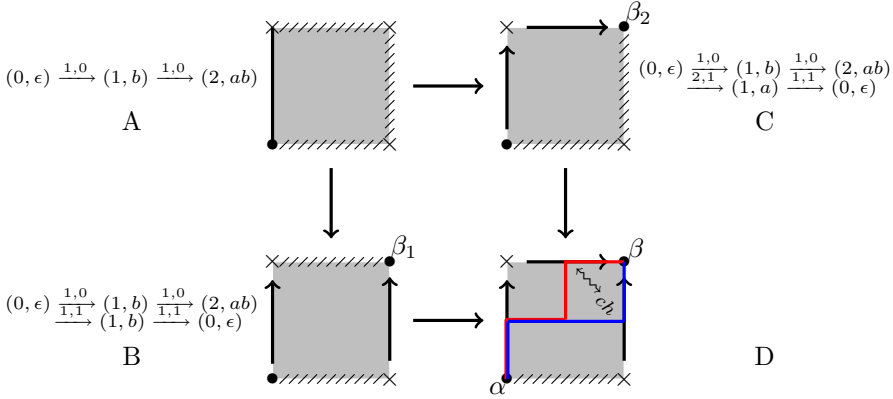
- $X_0 = \{(u, k) \mid u \in \mathcal{C}, k \leq l_u \wedge d_k^u = 0\} \sqcup \{\epsilon\}$,
- $X_n = \{(u, k) \mid u \in \mathcal{C}, k \leq l_u \wedge d_k^u = n\}$.

We quotient X_n by the smallest equivalence relation \sim (for inclusion) such that:

- for every u , $(u, 0) \sim \epsilon$,
- if $i : u \longrightarrow v \in \mathcal{C}$, and if $k \leq l_u, l_v$, then $(u, k) \sim (v, k)$.

We denote by Y_n the quotient X_n / \sim , and by $[u, k]$ the equivalence class of (u, k) modulo \sim .

At this stage, we still do not have the colimit because it is not possible to define the face maps. Let us consider the following example.



A , B and C are path shapes, and we would like to compute their pushout. The expected outcome is D , since we must identify the three squares by the previous construction. The problem is that the previous construction does not identify β_1 and β_2 . Those two must be identified because they are both the top right corner of the same square (after identification). We hence need to quotient a little more to be able to define the face maps, as follows. Define Z_n to be the quotient of Y_n by the smallest equivalence relation \approx such that if there are two sequences u_0, \dots, u_l and v_0, \dots, v_l such that:

- $[u_0, k] \approx [v_0, k]$,
- for every $0 \leq s \leq l$, $\alpha_{k+1+s}^{u_s} = \alpha_{k+1+s}^{v_s} = 1$,
- for every $0 \leq s < l$, $[u_s, k+s+1] \approx [u_{s+1}, k+s+1]$ and $[v_s, k+s+1] \approx [v_{s+1}, k+s+1]$,
- $(j_{k+1}^{u_0}; 1) \star \dots \star (j_{k+l+1}^{u_l}; 1) = (j_{k+1}^{v_0}; 1) \star \dots \star (j_{k+l+1}^{v_l}; 1)$,
then, $[u_l, k+l+1] \approx [v_l, k+l+1]$. $\text{col } D$ is the pHDA Z_N with the face maps being the smallest relations for inclusion such that:
 - if $\alpha_k^u = 0$, then $\partial_{j_k^u}^0(\langle u, k \rangle)$ is defined and is equal to $\langle u, k-1 \rangle$,
 - if $\alpha_{k+1}^u = 1$ then $\partial_{j_k^u}^1(\langle u, k \rangle)$ is defined and is equal to $\langle u, k+1 \rangle$,
 - $\partial_{j_1 < \dots < j_m}^{\beta_1, \dots, \beta_m} \circ \partial_{i_1 < \dots < i_n}^{\alpha_1, \dots, \alpha_n} \subseteq \partial_{k_1 < \dots < k_{n+m}}^{\gamma_1, \dots, \gamma_{n+m}}$, for $(k_1, \dots, k_{n+m}; \gamma_1, \dots, \gamma_{n+m}) = (i_1, \dots, i_n; \alpha_1, \dots, \alpha_n) \star (j_1, \dots, j_m; \beta_1, \dots, \beta_m)$.

The initial state is $\langle \epsilon \rangle$ and the labelling $\lambda : \text{col } D \longrightarrow L$ maps $\langle u, k \rangle$ to w_k^u .

Proposition 3. *colD is the colimit of D in \mathbf{pHDA}_L*

By **tree** we mean any pHDA that is the colimit of a diagram with values in path shapes. We denote by \mathbf{Tr}_L the full subcategory of trees.

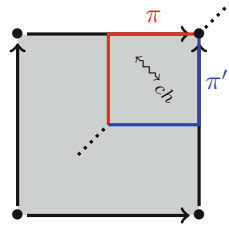
4.2 The Unique Path Properties of Trees

Failure of the Unicity of Paths. Let us consider the pushout square above again. In particular, the pHDA on the bottom-right corner is a tree, by definition. However, there are two paths from α to β (in red and blue). This actually comes from the fact that we needed to identify β_1 and β_2 to be able to define the face maps. This means that trees do not have the unique path property.

Confluent Homotopy. A careful reader may have observed that the only difference between the two previous paths is that some future faces are swapped. Actually, this is the only obstacle for the unicity of paths for trees: there is a unique path modulo equivalence of paths that permutes arrows of the form $\xrightarrow{-1}$. That is what we call **confluent homotopy**. This confluent homotopy will be defined by restricting the elementary homotopies of [12] to be of only one type out of the four possible, which means our notion of homotopy makes fewer paths equivalent than the one from [12].

We say that a path $\pi = x_0 \xrightarrow{j_1, \alpha_1} x_1 \xrightarrow{j_2, \alpha_2} \dots \xrightarrow{j_n, \alpha_n} x_n$ is **elementary confluent homotopic** to a path $\pi' = x'_0 \xrightarrow{j'_1, \alpha'_1} x'_1 \xrightarrow{j'_2, \alpha'_2} \dots \xrightarrow{j'_n, \alpha'_n} x'_n$, and denote by $\pi \rightsquigarrow_{ch} \pi'$, if and only if there are $0 < s < t \leq n$ such that:

- for all $k < s$ or $k \geq t$, $x_k = x'_k$,
- for all $k < s$ or $k > t$, $j_k = j'_k$ and $\alpha_k = \alpha'_k$,
- for all $s \leq k \leq t$, $\alpha_k = \alpha'_k = 1$,
- $(j_s, \alpha_s) \star \dots \star (j_t, \alpha_t) = (j'_s, \alpha'_s) \star \dots \star (j'_t, \alpha'_t)$.



We denote by \sim_{ch} , and call **confluent homotopy**, the reflexive transitive closure of \rightsquigarrow_{ch} .

Lemma 1. *If X is a tree, then for every element (of any dimension) x of X, there is exactly one path modulo confluent homotopy from the initial state to x.*

Shortcuts. The face maps of path shapes and of the colimits we computed in Sect. 4.1 are of a very particular form: we start by defining the ∂_j^α and we extend this definition to general $\partial_{j_1 < \dots < j_n}^{\alpha_1, \dots, \alpha_n}$. In a way, they are locally defined, and then extended to higher face maps. This means in particular that, in addition to having unique paths modulo confluent homotopy, they also do not have any ‘shortcut’. A possible shortcut can be defined as a generalisation of paths, in which we allow to make transitions that go, for example, from a point to a square or to a cube, not only to segments, a shortcut being such a possible shortcut which is not confluent homotopic to a path. Those shortcuts may occur in a

pHDA, even if it has the unique path property. Concretely, by **shortcut** we mean the following situation: the face $\partial_{i_1 < \dots < i_n}^{\alpha_1, \dots, \alpha_n}(x)$ is defined, but there is no sequence $(j_1; \beta_1) \star \dots \star (j_n; \beta_n) = (i_1 < \dots < i_n; \alpha_1, \dots, \alpha_n)$ such that $\partial_{j_n}^{\alpha_n} \circ \dots \circ \partial_{j_1}^{\alpha_1}(x)$ is defined. By local-definedness of the face maps:

Lemma 2. *Trees do not have any shortcuts.*

Trees. We say that a pHDA **has the unique path property modulo confluent homotopy** if it has no shortcut, and there is exactly one class of paths modulo confluent homotopy from the initial state to any state. Given such a pHDA X and an element x of X , by **depth of x** we mean the length of a path from the initial state to x in X . Since homotopic paths have the same length, this is uniquely defined. We deduce from the previous discussions that:

Proposition 4. *Trees have unique path property modulo confluent homotopy.*

In the following, we will prove the converse: trees, defined as colimits of path shapes are exactly those pHDA that have the unique path property modulo confluent homotopy. This will be done by proving that such a pHDA X is isomorphic to its unfolding. A question that occurs now is the following. Much as the general framework of [4], trees are colimits of paths. Everything tends to work well when those trees have a nice property, which we called **accessibility**, intuitively, that the colimit process do not ‘create’ paths. This property is actually deeply related to the unicity of paths. Since this unicity fails in the case of pHDA, accessibility fails too. However, an accessibility modulo confluent homotopy holds: the colimit process in pHDA do not create confluent homotopy classes of paths.

4.3 Trees Are Unfoldings

We are now constructing our unfolding $U(X)$ of a pHDA X by giving an explicit definition, similar to [6, 11], and proving that this is a tree. We will prove that there is a covering $\text{unf}_X : U(X) \rightarrow X$, which in particular means that the unfolding $U(X)$ is **PS_L**-bisimilar (in the general sense of [17]) to X , and that this covering is actually an isomorphism when X has the unique path property modulo confluent homotopy.

Unfolding of a pHDA. Let us start with a few notations. Given a path $\pi = x_0 \xrightarrow{j_1, \alpha_1} x_1 \xrightarrow{j_2, \alpha_2} \dots \xrightarrow{j_n, \alpha_n} x_n$ we note $e(\pi) = x_n$, $l(\pi) = n$ and $\pi_{-k} = x_0 \xrightarrow{j_1, \alpha_1} x_1 \xrightarrow{j_2, \alpha_2} \dots \xrightarrow{j_{n-k}, \alpha_{n-k}} x_{n-k}$. Given a pHDA X , its **unfolding** is the following pHDA:

- $U(X)_n$ is the set of equivalence classes $[\pi]$ of paths modulo confluent homotopy, such that $e(\pi)$ is of dimension n ,
- the face maps are the smallest relations for inclusion such that:
 - $\partial_i^1(\alpha) = [\pi \xrightarrow{i,1} \partial_i^1(e(\pi))]$, for any $\pi \in \alpha$ such that $\partial_i^1(e(\pi))$ is defined,
 - $\partial_i^0(\alpha) = [\pi_{-1}]$ for any $\pi \in \alpha$ such that $\pi = \pi_{-1} \xrightarrow{i,0} e(\pi)$,

- $\partial_{j_1 < \dots < j_m}^{\beta_1, \dots, \beta_m} \circ \partial_{i_1 < \dots < i_n}^{\alpha_1, \dots, \alpha_n} \subseteq \partial_{k_1 < \dots < k_{n+m}}^{\gamma_1, \dots, \gamma_{n+m}}$, for $(k_1, \dots, k_{n+m}; \gamma_1, \dots, \gamma_{n+m}) = (i_1, \dots, i_n; \alpha_1, \dots, \alpha_n) \star (j_1, \dots, j_m; \beta_1, \dots, \beta_m)$.
- the initial state is $[i]$,
- the labelling is given by $\lambda(\alpha) = \lambda(e(\pi))$ for $\pi \in \alpha$.

Following ideas from [4] again, the unfolding can be seen as the glueing of all possible executions of a system, but with care needed to handle confluent homotopy. Concretely:

Proposition 5. *The unfolding of a pHDA is a tree.*

We can also define $\text{unf}_X : U(X) \longrightarrow X$ as the function that maps $[\pi]$ to $e(\pi)$.

Proposition 6. *unf_X is a covering, and so, $U(X)$ is \mathbf{PS}_L -bisimilar to X .*

The Unique Path Property Characterises Trees. When X has exactly one class of paths modulo confluent homotopy from the initial state to any state, it is possible to define a function $\eta_X : X \longrightarrow U(X)$ that maps any element x of X to the unique confluent homotopy class to x . When furthermore X does not have shortcuts, then η is actually a morphism of pHDA.

Proposition 7. *When X has the unique path property modulo confluent homotopy, then η_X is the inverse of unf_X . In particular, X is a tree.*

Together with Proposition 4, this implies the following:

Theorem 2. *Trees are exactly the pHDA that have the unique path property modulo confluent homotopy.*

Another consequence is that this isomorphism η_X is actually natural (in the categorical sense) and is part of an adjunction, which implies that trees form a coreflective subcategory of pHDA:

Corollary 1. *U extends to a functor, which is the right adjoint of the embedding $\iota : \mathbf{Tr}_L \longrightarrow \mathbf{pHDA}_L$. Furthermore, this is a coreflection.*

5 Cofibrant Objects

Cofibrant objects are another type of ‘simple objects’, coming from homotopy theory, more particularly the language of model categories from [24]. Those cofibrant objects are those whose unique morphism from the initial object is a cofibration. Intuitively (intuition which holds at least in cofibrantly generated model structures [15]), this means that cofibrant objects are those objects constructed from ‘nothing’, using only very basic constructions (generators of cofibrations). In the case of the classical model structure on topological spaces (Kan-Quillen), those spaces are those constructed from the empty space by adding ‘cells’, which produces what is called CW-complexes. In this section, we want to mimic this idea with trees: trees are those pHDA constructed from an initial state by only extending paths. We also want to emphasize that much as CW-complexes gives a kind of homotopy type of a space, trees gives a concurrency type of a pHDA, in the sense that there is a canonical way to produce an equivalent cofibrant object out of any object, which is called the **cofibrant replacement** in homotopy theory. In concurrency theory, this is the unfolding.

5.1 Cofibrant Objects in $\mathbf{pHDA}_{\mathbf{L}}$

Following the language of model structures from [24], we say that a \mathbf{pHDA} X is **cofibrant** if for every $\mathbf{PS}_{\mathbf{L}}$ -open morphism $f : Y \longrightarrow Z$ and every morphism $g : X \longrightarrow Z$, there is a morphism $h : X \longrightarrow Y$, such that $f \circ h = g$. That is, a partial HDA X is cofibrant if and only if every $\mathbf{PS}_{\mathbf{L}}$ -open morphism has the right lifting property with respect to the unique morphism from $*$ to X .

$$\begin{array}{ccc} * & \xrightarrow{!} & Y \\ \downarrow ! & \nearrow h & \downarrow f \\ X & \xrightarrow{g} & Z \end{array}$$

5.2 Cofibrant Objects Are Exactly Trees

In this section, we would like to prove the following:

Theorem 3. *The cofibrant objects are exactly trees.*

Let us start by giving the idea of the proof of the fact that cofibrant objects are trees. By Proposition 6, unf_X is a covering, so is open. This means that for every cofibrant object X , there is a morphism $h : X \longrightarrow U(X)$ such that $\text{unf}_X \circ h = \text{id}_X$, that is, X is a retract of its unfolding. Since we know that the unfolding is a tree by Proposition 5, it is enough to observe the following:

$$\begin{array}{ccc} * & \xrightarrow{!} & U(X) \\ \downarrow ! & \nearrow h & \downarrow \text{unf} \\ X & \xrightarrow{\text{id}_X} & X \end{array}$$

Lemma 3. *A retract of a tree is a tree.*

Intuitively, a \mathbf{pHDA} is the retract of a tree only when it is obtain by retracting branches. This can only produce a tree. For the converse:

Proposition 8. *A tree is a cofibrant object. Furthermore, if $f : Y \longrightarrow Z$ is a covering, then the lift $h : X \longrightarrow Y$ is unique.*

The lift h is constructed by induction as follows. We define X_n as the restriction of X to elements whose depth is smaller than n , and the face maps $\partial_{j_1 < \dots < j_m}^{\alpha_1, \dots, \alpha_m}(x)$ are defined if and only if $\partial_{j_1 < \dots < j_m}^{\alpha_1, \dots, \alpha_m}(x)$ is defined in X and belongs to X_n . We then construct $h_n : X_n \longrightarrow Y$ using the unique path property modulo confluent homotopy, in a natural way (in the categorical meaning), i.e., such that $h_n \circ \kappa_n = h_{n-1}$, where $\kappa_n : X_{n-1} \longrightarrow X_n$ is the inclusion. h is then the inductive limit of those h_n . This proof can be seen as a small object argument.

5.3 The Unfolding Is Universal

As an application of the previous theorem, we would like to prove that the unfolding is universal. As in the case of covering spaces in algebraic topology, a covering corresponds to a partial unrolling of a system, in the sense that we can unroll some loops or even partially unroll a loop (imagine for example executing a few steps of a while-loop). In this sense, we can describe the fact that a covering unrolls more than another one, and that, an unfolding is a complete unrolling: since the domain is a tree, it is impossible to unroll more. Actually, much as the

topological and the groupoidal cases (see [18] for example), unfoldings are the only such maximal unrollings among coverings: they are initial among coverings, that is why we call them ‘universal’. In a way, this says that our definition of unfolding is the only reasonable one. Concretely, we say that a \mathbf{PS}_L -covering is **universal** if its domain is a tree.

Corollary 2. *If $f : Y \longrightarrow X$ is a universal covering, then for every covering $g : Z \longrightarrow X$ there is a unique map $h : Y \longrightarrow Z$ such that $f = g \circ h$. Furthermore, h is itself a covering. Consequently, the universal covering is unique up-to isomorphism, and is given by the unfolding.*

This whole story is similar to the universal covering of a topological space: just replace pHDA by spaces and trees by simply-connected spaces [2].

6 Conclusion and Future Work

In this paper, we have given a cleaner definition of partial precubical sets and partial Higher Dimensional Automata, as they really correspond to collections of cubes with missing faces. From this categorical definition, we derived that pHDA can be completed, giving rise to a geometric realisation. We also describe the first premisses of a homotopy theory of the concurrency of pHDA where the cofibrant objects are trees, and replacement is the unfolding. As a future work, we could look at wider class of paths, typically allowing shortcuts as paths, or introducing general homotopies in the path category, which is possible because we can encode those inside the category of pHDA. Another direction would be to continue the description of this homotopy theory, to see if it corresponds to some kind of Quillen’s model structure, or at least to some weaker version (e.g., category of cofibrant objects).

References

1. Bednarczyk, M.A.: Categories of asynchronous systems. Ph.D. thesis, University of Sussex (1987)
2. tom Dieck, T.: Algebraic Topology. Textbooks in Mathematics. European Mathematical Society, Zürich (2008)
3. Dubut, J.: Directed homotopy and homology theories for geometric models of true concurrency. Ph.D. thesis, ENS Paris-Saclay (2017)
4. Dubut, J., Goubault, E., Goubault-Larrecq, J.: Bisimulations and unfolding in P-accessible categorical models. In: Proceedings of the 27th International Conference on Concurrency Theory (CONCUR 2016). Leibniz International Proceedings in Informatics (LIPIcs), vol. 59, pp. 1–14. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2016)
5. Fahrenberg, U.: A category of higher-dimensional automata. In: Sassone, V. (ed.) FoSSaCS 2005. LNCS, vol. 3441, pp. 187–201. Springer, Heidelberg (2005). https://doi.org/10.1007/978-3-540-31982-5_12
6. Fahrenberg, U., Legay, A.: History-preserving bisimilarity for higher-dimensional automata via open maps. Electron. Notes Theor. Comput. Sci. **298**, 165–178 (2013)

7. Fahrenberg, U., Legay, A.: Partial higher-dimensional automata. In: CALCO 2015, pp. 101–115 (2015)
8. Fajstrup, L.: Dipaths and dihomotopies in a cubical complex. *Adv. Appl. Math.* **35**(2), 188–206 (2005)
9. Fajstrup, L., Goubault, E., Haucourt, E., Mimram, S., Raussen, M.: *Directed Algebraic Topology and Concurrency*. Springer, Cham (2016). <https://doi.org/10.1007/978-3-319-15398-8>
10. Gaucher, P.: Towards a homotopy theory of higher dimensional transition systems. *Theory Appl. Categ.* **25**, 295–341 (2011)
11. van Glabbeek, R.J.: Bisimulations for higher dimensional automata, June 1991. <http://theory.stanford.edu/~rvg/hda>
12. van Glabbeek, R.J.: On the expresiveness of higher dimensional automata: (extended abstract). *Electron. Notes Theor. Comput. Sci.* **128**(2), 5–34 (2005)
13. Goubault, E.: *Géométrie du parallélisme*. Ph.D. thesis, Ecole Polytechnique (1995)
14. Grandis, M.: *Directed Algebraic Topology: Models of Non-Reversible Worlds*. New Mathematical Monographs, vol. 13. Cambridge University Press, Cambridge (2009)
15. Hirschhorn, P.S.: *Model Categories and Their Localizations*. Mathematical Surveys and Monographs, vol. 99. American Mathematical Society, Providence (2003)
16. Jacobs, B.: *Introduction to Coalgebra: Towards Mathematics of States and Observation*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, New York (2016)
17. Joyal, A., Nielsen, M., Winskel, G.: Bisimulation from open maps. *Inf. Comput.* **127**(2), 164–185 (1996)
18. May, J.P.: *A Concise Course in Algebraic Topology*. Chicago Lectures in Mathematics. University of Chicago Press, Chicago (1999)
19. Milner, R. (ed.): *A Calculus of Communicating Systems*. LNCS, vol. 92. Springer, Heidelberg (1980). <https://doi.org/10.1007/3-540-10235-3>
20. Niefeld, S.: Lax presheaves and exponentiability. *Theory Appl. Categ.* **24**(12), 288–301 (2010)
21. Nielsen, M., Plotkin, G., Winskel, G.: Petri nets, event structures and domains, part I. *Theor. Comput. Sci.* **13**(1), 85–108 (1981)
22. Nielsen, M., Sassone, V., Winskel, G.: Relationships between models of concurrency. In: de Bakker, J.W., de Roever, W.-P., Rozenberg, G. (eds.) *REX 1993*. LNCS, vol. 803, pp. 425–476. Springer, Heidelberg (1994). https://doi.org/10.1007/3-540-58043-3_25
23. Pratt, V.: Modeling concurrency with geometry. In: *Proceedings of the 18th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL)*, pp. 311–322, January 1991
24. Quillen, D.G.: *Homotopical Algebra*. LNM, vol. 43. Springer, Heidelberg (1967). <https://doi.org/10.1007/BFb0097438>
25. Shields, M.W.: Concurrent machines. *Comput. J.* **28**(5), 449–465 (1985)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

