Modal Logic over Higher Dimensional Automata[⋆]

Cristian Prisacariu

Dept. of Informatics, Univ. of Oslo, P.O. Box 1080 Blindern, N-0316 Oslo, Norway.
cristi@ifi.uio.no

Abstract. Higher dimensional automata (HDA) are a model of concurrency that can express most of the traditional partial order models like Mazurkiewicz traces, pomsets, event structures, or Petri nets. Modal logics, interpreted over Kripke structures, are the logics for reasoning about sequential behavior and interleaved concurrency. Modal logic is a well behaved subset of first-order logic; many variants of modal logic are decidable. However, there are no modal-like logics for the more expressive HDA models. In this paper we introduce and investigate a modal logic over $HDAs$ which incorporates two modalities for reasoning about "during" and "after". We prove that this general higher dimensional modal logic ($HDML$) is decidable and we define a complete axiomatic system for it. We also show how, when the HDA model is restricted to Kripke structures, a syntactic restriction of $HDML$ becomes the standard modal logic. Then we isolate the class of $HDAs$ that encode Mazurkiewicz traces and show how $HDML$ can be restricted to LTrL (the linear time temporal logic over Mazurkiewicz traces).

1 Introduction

Higher dimensional automata ($HDAs$) are a general formalism for modeling concurrent systems [14, 21]. In this formalism concurrent systems can be modeled at different levels of abstraction, not only as all possible interleavings of their concurrent actions. $HDAs$ can model concurrent systems at any granularity level and make no assumptions about the durations of the actions. Moreover, $HDAs$ are not constrained to only before-after modeling and expose explicitly the choices in the system. It is a known issue in concurrency models that the combination of causality, concurrency, and choice is difficult; in this respect, $HDAs$ and Chu spaces [15] do a fairly good job [17].

Higher dimensional automata are more expressive than most of the models based on partial orders or on interleavings (e.g., Petri nets and the related Mazurkiewicz traces, or the more general partial order models like pomsets or event structures). Therefore, one only needs to find the right class of $HDAs$ in order to get the desired models of concurrency.

Work has been done on defining temporal logics over Mazurkiewicz traces [9] and strong results like decidability and expressive completeness are known [5, 20]. For general partial orders, temporal logics usually become undecidable [2]. For the more expressive event structures there are fewer works; a modal logic is investigated in [6].

There is hardly any work on logics for higher dimensional automata [17] and, as far as we know, there is no work on *modal logics for* $HDAs$. In practice, one is more comfortable with modal logics, like temporal logics or dynamic logics, because these are generally decidable (as opposed to full first-order logic, which is undecidable).

⋆ The appendix is for reviewing purpose only and should not be regarded as part of the paper; it contains the full proofs for all the results.

[0] A technical report of this paper, containing proofs and more explanations, appeared as [18].

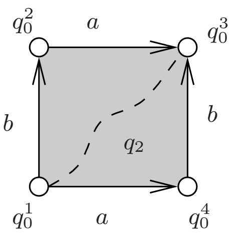

Fig. 1. Example of a HDA with two concurrent events.

That is why in this paper we introduce and develop a logic in the style of standard modal logic. This logic has $HDAs$ as models, hence, the name *higher dimensional modal logic* ($HDML$). This is our basic language to talk about general models of concurrent systems. For this basic logic we prove decidability using a filtration argument. Also, we provide an axiomatic system and prove it is sound and complete for the higher dimensional automata. $HDML$ in its basic variant is shown to become standard modal logic when the language and the higher dimensional models are restricted in a certain way.

$HDML$ contrasts with standard temporal/modal logics in the fact that $HDML$ can reason about *what holds "during" some concurrent events are executing*. The close related logic for distributed transition systems of [7] is in the same style of reasoning only about what holds "after" some concurrent events have finished executing. As we show in the examples section, the "after" logics can be encoded in $HDML$, hence also the logic of [7].

The other purpose of this work is to provide a general framework for reasoning about concurrent systems at any level of abstraction and granularity, accounting also for choices and independence of actions. Thus, the purpose of the examples in Section 3 is to show that studying $HDML$, and particular variants of it, is fruitful for analyzing concurrent systems and their logics. In this respect we study variants of higher dimensional modal logic inspired by temporal logic and dynamic logic. Already in Section 3.2 we add to the basic language an *Until* operator, in the style of temporal logics. We show how this variant of $HDML$, when interpreted over the class of $HDAs$ corresponding to Kripke structures, can be particularized to LTL [11]. A second variant, in Section 3.3, decorates the $HDML$ modalities with labels. This multi-modal variant of $HDML$ together with the *Until* operator, when interpreted over the class of $HDAs$ that encodes Mazurkiewicz traces, becomes LTrL [20] (the linear time temporal logic over Mazurkiewicz traces).

2 Modal Logic over Higher Dimensional Automata

In this section we define a higher dimensional automaton (HDA) following the definition and terminology of [17,21]. Afterwards, we propose *higher dimensional modal logic* ($HDML$) for reasoning about concurrent systems modeled as $HDAs$. The semantic interpretation of the language is defined in terms of $HDAs$ (i.e., the $HDAs$, with a valuation function attached, are the models we propose for $HDML$).

For an intuitive understanding of the HDA model consider the standard example [17, 21] pictured in Figure 1. It represents a HDA that models two concurrent events which are labeled by a and b (one might have the same label a for both events). The HDA has four states, q_0^1 to q_0^4, and four transitions between them. This would be the standard picture for interleaving, but in the case of HDA there is also a square q_2. Traversing through the interior of the square means that both events are executing. When traversing on the lower transition means that event one is executing but event two has not started yet, whereas, when traversing through the upper transition it means that event one is executing and event two has finished already. In the states there is no event executing, in particular, in state q_0^3 both events have finished, whereas in state q_0^1 no event has started yet.

In the same manner, $HDAs$ allow to represent three concurrent events through a cube, or more events through hypercubes. Causality of events is modeled by sticking such hypercubes one after the other. For our example, if we omit the interior of the square (i.e., the grey q_2 is removed) we are left with a description of a system where there is the choice between two sequences of two events, i.e., $a;b+b;a$.

Definition 2.1 (higher dimensional automata). *A* cubical set $H = (Q,\overline{s},\overline{t})$ *is formed of a family of sets* $Q = \bigcup_{n=0}^{\infty} Q_n$ *with all sets* Q_n *disjoint, and for each* n*, a family of maps* $s_i, t_i : Q_n \to Q_{n-1}$ *with* $1 \le i \le n$ *which respect the following* cubical laws*:*

$$\alpha_i \circ \beta_j = \beta_{j-1} \circ \alpha_i, \quad 1 \le i < j \le n \text{ and } \alpha,\beta \in \{s,t\}. \tag{1}$$

In H*, the* $\overline{s}$ *and* $\overline{t}$ *denote the collection of all the maps from all the families (i.e., for all* n*). A* higher dimensional structure $(Q,\overline{s},\overline{t},l)$ *over an alphabet* Σ *is a cubical set together with a* labeling function $l : Q_1 \to \Sigma$ *which respects* $l(s_i(q)) = l(t_i(q))$ *for all* $q \in Q_2$ *and* $i \in \{1,2\}$.[1] *A* higher dimensional automaton $(Q,\overline{s},\overline{t},l,I,F)$ *is a higher dimensional structure with two designated sets of* initial *and* final cells $I \subseteq Q_0$ *and* $F \subseteq Q_0$.

We call the elements of Q_0, Q_1, Q_2, Q_3 respectively *states*, *transitions*, *squares*, and *cubes*, whereas the general elements of Q_n are called n-dimensional cubes (or hypercubes). We call generically an element of Q a *cell* (also known as n-cell). For a transition $q \in Q_1$ the $s_1(q)$ and $t_1(q)$ represent respectively its source and its target cells (which are *states* from Q_0 in this case). Similarly for a general cell $q \in Q_n$ there are n source cells and n target cells all of dimension $n-1$. Intuitively, an n-dimensional cell q represents a configuration of a concurrent system in which n events are performed at the same time, i.e., concurrently. A source cell $s_i(q)$ represents the configuration of the system before the starting of the i^{th} event, whereas the target cell $t_i(q)$ represents the configuration of the system immediately after the termination of the i^{th} event. A transition of Q_1 represents a configuration of the system in which a single event is performed.

The cubical laws account for the geometry (concurrency) of the $HDAs$; there are four kinds of cubical laws depending on the instantiation of α and β. For the example of Figure 1 consider the cubical law where α is instantiated to t and β to s, and $i=1$ and $j=2$: $t_1(s_2(q_2)) = s_1(t_1(q_2))$. In the left hand side, the second source cell of q_2 is, in this case, the transition $s_2(q_2) = q_1^1 = (q_0^1, q_0^2)$ and the first target cell of q_1^1 is q_0^2 (the only target cell because $s_2(q_2) \in Q_1$); this must be the same cell when taking the right hand side of the cubical law, i.e., the first target cell is $t_1(q_2) = q_1^2 = (q_0^2, q_0^3)$ and the first source of q_1^2 is q_0^2.

We propose the language of higher dimensional modal logic for talking about concurrent systems. $HDML$ follows the tradition and style of standard modal languages [3].

[1] Later, in Definition 3.12, the labeling is extended naturally to all cells.

$\mathcal{H}, q \models \phi$	iff $\phi \in \mathcal{V}(q)$.
$\mathcal{H}, q \not\models \bot$	
$\mathcal{H}, q \models \varphi_1 \rightarrow \varphi_2$	iff when $\mathcal{H}, q \models \varphi_1$ then $\mathcal{H}, q \models \varphi_2$.
$\mathcal{H}, q \models \{\}\varphi$	iff assuming $q \in Q_n$ for some n, $\exists q' \in Q_{n+1}$ s.t. $s_i(q') = q$ for some $1 \leq i \leq n$, and $\mathcal{H}, q' \models \varphi$.
$\mathcal{H}, q \models \langle\rangle\varphi$	iff assuming $q \in Q_n$ for some n, $\exists q' \in Q_{n-1}$ s.t. $t_i(q) = q'$ for some $1 \leq i \leq n$, and $\mathcal{H}, q' \models \varphi$.

Table 1. Semantics for $HDML$.

Definition 2.2 (higher dimensional modal logic). *A formula φ in higher dimensional modal logic is constructed using the grammar below, from a set Φ_B of atomic propositions, with $\phi \in \Phi_B$, which are combined using the propositional symbols $\bot$ and $\rightarrow$ (from which all other standard propositional operations are generated), and using the modalities $\{\}$ and $\langle\rangle$.*

$$\varphi := \phi \mid \bot \mid \varphi \rightarrow \varphi \mid \{\}\varphi \mid \langle\rangle\varphi$$

We call $\{\}$ the *during modality* and $\langle\rangle$ the *terminate modality*. The intuitive reading of $\{\}\varphi$ is: "pick some event from the ones currently not running (must exist at least one not running) and start it; in the new configuration of the system (during which, one more event is concurrently executing) the formula φ must hold". The intuitive reading of $\langle\rangle\varphi$ is: "pick some event from the ones currently running concurrently (must exist one running) and terminate it; in the new configuration of the system the formula φ must hold". This intuition is formalized in the semantics of $HDML$.

The choice of our notation is biased by the intuitive usage of these modalities where the terminate modality talks about what happens after some event is terminated; in this respect being similar to the standard diamond modality of dynamic logic. Later, in Section 3.3, these modalities are decorated with labels. The during modality talks about what happens during the execution of some event and hence we adopt the notation of Pratt [12].

The models of $HDML$ are higher dimensional structures together with a valuation function $\mathcal{V} : Q \rightarrow 2^{\Phi_B}$ which associates a set of atomic propositions to each cell (of any dimension). This means that $\mathcal{V}$ assigns some propositions to each state of dimension 0, to each transition of dimension 1, to each square of dimension 2, to each cube of dimension 3, etc. Denote a model of $HDML$ by $\mathcal{H} = (Q, s, t, l, \mathcal{V})$. A $HDML$ formula is evaluated in a cell of such a model $\mathcal{H}$.

One may see the $HDML$ models as divided into *levels*, each level increasing the concurrency complexity of the system; i.e., level Q_n increases the complexity compared to level Q_{n-1} by adding one more event (to have n events executing concurrently instead of $n-1$). The levels are linked together through the s_i and t_i maps. With this view in mind the during and terminate modalities should be understood as jumping from one level to the other; the $\{\}$ modality jumps one level up, whereas the $\langle\rangle$ modality jumps one level down.

Definition 2.3 (satisfiability). *Table 1 defines recursively the satisfaction relation $\models$ of a formula φ w.r.t. a model $\mathcal{H}$ in a particular n-cell q (for some arbitrary n); denote this as $\mathcal{H}, q \models \varphi$. The notions of satisfiability and validity are defined as usual.*

Both modalities have an existential flavor. In particular note that $\mathcal{H}, q_0 \not\models \langle\rangle\varphi$, for $q_0 \in Q_0$ a state, because there is no event executing in a state, and thus no event can be terminated. Similarly, for the during modality, $\mathcal{H}, q_n \not\models \{\}\varphi$ for any n-cell $q_n \in Q_n$

when all sets Q_k, with $n < k$, are empty (i.e., the family of sets Q is bounded by n). This says that there can be at most n events running at the same time, and when reaching this limit one cannot start another event and therefore $\{\}\varphi$ cannot be satisfied.

The universal correspondents of $\{\}$ and $\langle\rangle$ are defined in the standard style of modal logic. We denote these modalities by respectively $\{\!\![\,]\!\!\}\varphi$ and $[\,]\varphi$. The intuitive reading of $[\,]\varphi$ is: "pick any of the events currently running concurrently and after terminating it, φ must hold in the new configuration of the system". Note that this modality holds trivially for any state $q_0 \in Q_0$, i.e., $\mathcal{H}, q_0 \models [\,]\varphi$.

2.1 Decidability and completeness

In the rest of this section we prove that satisfiability for *HDML* is decidable using the filtration technique [3]. Then we give an axiomatic system for *HDML* and prove its soundness and completeness. Completeness is based on constructing canonical models.

The filtration for the states is the same as in the standard modal logic, but for cells of dimension at least 1 we need to take care that the maps t and s in the filtration model remain maps and that they respect the cubical laws so that the filtration is still a *HDML* model. This can be done, but the filtration model is bigger than what is obtained in the case of standard modal logic. On top, the proof of the small model property (Theorem 2.10) is more involved due to the complexities of the definition of filtration given in Definition 2.5.

Definition 2.4 (subformula closure). *The* subformula closure *of a formula φ is the set of formulas $\mathcal{C}(\varphi)$ defined recursively as:*

$$\begin{aligned}
\mathcal{C}(\phi) &\triangleq \{\phi\}, \text{ for } \phi \in \Phi_B\\
\mathcal{C}(\varphi_1 \to \varphi_2) &\triangleq \{\varphi_1 \to \varphi_2\} \cup \mathcal{C}(\varphi_1) \cup \mathcal{C}(\varphi_2)\\
\mathcal{C}(\{\}\varphi) &\triangleq \{\{\}\varphi\} \cup \mathcal{C}(\varphi)\\
\mathcal{C}(\langle\rangle\varphi) &\triangleq \{\langle\rangle\varphi\} \cup \mathcal{C}(\varphi)
\end{aligned}$$

The *size* of a formula (denoted $|\varphi|$) is calculated by summing the number of Boolean and modal symbols with the number of atomic propositions and $\bot$ symbols that appear in the formula. (All instances of a symbol are counted.) The size of the subformula closure of a formula φ is linear in the size of the formula, $|\mathcal{C}(\varphi)| \leq |\varphi|$.

Definition 2.5 (filtration). *Given a formula φ, we define below a relation $\equiv$ (which is easily proven to be an equivalence relation) over the cells of a higher dimensional structure $\mathcal{H}$, where $q, q' \in Q_i$, for some $i \in \mathbb{N}$:*

$$q \equiv q' \text{ iff for any } \psi \in \mathcal{C}(\varphi) \text{ then } (\mathcal{H}, q \models \psi \text{ iff } \mathcal{H}, q' \models \psi).$$

A filtration model *$\mathcal{H}^f$ of $\mathcal{H}$ through the closure set $\mathcal{C}(\varphi)$ is a structure $(Q^f, s^f, t^f, l^f, \mathcal{V}^f)$:*

$$\begin{aligned}
Q^f_n &\triangleq \{[q_n] \mid q_n \in Q_n\}, \text{ where } [q_n] \text{ is}\\
[q_0] &\triangleq \{q' \mid q_0 \equiv q'\} \text{ when } q_0 \in Q_0, \text{ otherwise,}\\
[q_n] &\triangleq \{q' \mid q_n \equiv q' \wedge t_i(q') \in [p_i] \wedge s_i(q') \in [p'_i]\\
&\qquad \text{for all } 1 \leq i \leq n \text{ and for some fixed } [p_i], [p'_i] \in Q^f_{n-1}\}.\\
s^f_i([q_n]) &\triangleq [q_{n-1}] \text{ iff for all } p \in [q_n], s_i(p) \in [q_{n-1}].\\
t^f_i([q_n]) &\triangleq [q_{n-1}] \text{ iff for all } p \in [q_n], t_i(p) \in [q_{n-1}].\\
\mathcal{V}^f([q]) &\triangleq \mathcal{V}(q).
\end{aligned}$$

Proposition 2.6 (filtration is a model). *The filtration $\mathcal{H}^f$ of a model $\mathcal{H}$ through a closure set $\mathcal{C}(\varphi)$ is a higher dimensional structure (i.e., is still a $HDML$ model).*

Proof (sketch). Essentially, the proof amounts to showing that the definitions of s_i^f and t_i^f are that of *maps* (as required in a higher dimensional structure) and that they respect the *cubical laws* (see full proof as Lemma A.3).

Lemma 2.7 (sizes of filtration sets). *Each set Q_n^f of the filtration $\mathcal{H}^f$ obtained in Definition 2.5 has finite size which depends on the size of the formula φ used in the filtration; more precisely each Q_n^f is bounded from above by $2^{|\varphi|\cdot N}$ where $N = n! \cdot \sum_{k=0}^{n} \frac{2^k}{(n-k)!}$.*

Proof. The case for 0 is simple as the number of equivalence classes of Q_0 can be maximum the number of subsets of the subformula closure $\mathcal{C}(\varphi)$ which is $2^{|\varphi|}$.

The case for $n = 1$ is based on the size of Q_0^f. Each of the $2^{|\varphi|}$ equivalence classes in which Q_1^f can be divided may have infinitely many cells. Any such equivalence class can still be broken into smaller subsets depending on the maps t_1 and s_1. Because t_1 can have outcome in any of the $[q_0] \in Q_0^f$, we get a first split into $2^{|\varphi|}$ subdivisions. For each of these we can still split it into $2^{|\varphi|}$ more subdivisions because of s_1. We thus get a maximum of $2^{|\varphi|} \cdot (2^{|\varphi|})^{2\cdot 1}$. For the general case of n we need to consider all maps t_i, s_i, that means $2 \cdot n$ maps. For each of these maps we split the $2^{|\varphi|}$ possible initial equivalence classes according to the size of O_{n-1}^f. Thus we get a maximum of $2^{|\varphi|} \cdot (|Q_{n-1}^f|)^{2\cdot n}$ subdivisions. Calculating this series gives the bound on the size of Q_n^f as being $2^{|\varphi|\cdot N}$ where $N = n! \cdot \sum_{k=0}^{n} \frac{2^k}{(n-k)!}$.

Lemma 2.8 (filtration lemma). *Let $\mathcal{H}^f$ be the filtration of $\mathcal{H}$ through the closure set $\mathcal{C}(\varphi)$, as in Definition 2.5. For any formula $\psi \in \mathcal{C}(\varphi)$ and any cell $q \in \mathcal{H}$, we have $\mathcal{H}, q \models \psi$ iff $\mathcal{H}^f, [q] \models \psi$.*

We define two *degrees of concurrency* of a formula φ: the *upwards concurrency* (denoted $|\varphi|_{uc}$) and *downwards concurrency* (denoted $|\varphi|_{dc}$). The degree of upwards concurrency counts the maximum number of nestings of the during modality $\{\}$ that are not compensated by a $\langle\rangle$ modality. (E.g., the formula $\{\}\{\}\phi \vee \{\}\phi'$ has the degree of upwards concurrency equal to 2, the same as $\{\}\langle\rangle\{\}\{\}\phi$.) The formal definition of $|\ |_{uc}$ is:

$$\begin{aligned}
|\bot|_{uc} &\triangleq |\phi|_{uc} \triangleq 0, \text{ for } \phi \in \Phi_B\\
|\varphi_1 \rightarrow \varphi_2|_{uc} &\triangleq max(|\varphi_1|_{uc}, |\varphi_2|_{uc})\\
|\{\}\varphi|_{uc} &\triangleq 1 + |\varphi|_{uc}\\
|\langle\rangle\varphi|_{uc} &\triangleq max(0, |\varphi|_{uc} - 1)
\end{aligned}$$

The definition of the degree of downwards concurrency $|\ |_{dc}$ is symmetric to the one above in the two modalities; i.e., interchange the modalities in the last two lines. Note that $|\varphi|_{uc} + |\varphi|_{dc} \leq |\varphi|$. The next result offers a safe reduction of a model where we remove all cells which have dimension greater than some constant depending on the formula of interest.

Lemma 2.9 (concurrency boundedness). *If a $HDML$ formula φ is satisfiable, $\mathcal{H}, q \models \varphi$ with $q \in Q_k$, then it exists a model with all the sets Q_m, with $m > |\varphi|_{uc} + k$, empty, which satisfies the formula.*

Notation: The formula $\langle\rangle\phi \wedge \langle\rangle\neg\phi$ expresses that there can be terminated at least two different events (in other words, the cell in which the formula is evaluated to true has dimension at least two). Similarly the formula $\langle\rangle(\phi \wedge \neg\phi') \wedge \langle\rangle(\neg\phi \wedge \neg\phi') \wedge \langle\rangle(\neg\phi \wedge \phi')$ says that there are at least three events that can be terminated. For each $i \in \mathbb{N}^*$ one can write such a formula to say that there are at least i events that can be terminated. Denote such a formula by $\langle\rangle i$. Also define $\langle\rangle^i\varphi$ as i applications of the $\langle\rangle$ modality to φ (i.e., $\langle\rangle \ldots \langle\rangle\varphi$ where $\langle\rangle$ appears i times). Similar, for the during modality denote $\{\}i$ the formula that can start i different events, and by $\{\}^i\varphi$ the i applications of $\{\}$ to φ.

Theorem 2.10 (small model property). *If a HDML formula φ is satisfiable then it is satisfiable on a finite model with no more than $\sum_{n=0}^{|\varphi|} 2^{|\varphi|\cdot N}$ cells where $N = n! \cdot \sum_{k=0}^{n} \frac{2^k}{(n-k)!}$.*

Proof. Note first that it is easy to prove that any formula $\langle\rangle i \rightarrow \langle\rangle^i\top$ is valid, for any $i \in \mathbb{N}^*$. Because of this, the downwards concurrency measure for a formula may be misleading as $|\langle\rangle i|_{dc} = 1$ whereas $|\langle\rangle^i\top|_{dc} = i$. On the other hand the dimension $|\langle\rangle i|$ grows faster than linear with i. It is easy to see that $|\langle\rangle i| > |\langle\rangle^i\top|_{dc}$. If our formula φ has subformulas of the kind $|\langle\rangle i|_{dc}$ then the measure $|\varphi|_{dc}$ must be adjusted accordingly. In any case, it is clear that even after adjustment $|\varphi|_{dc} \leq |\varphi| - |\varphi|_{uc}$.

Assume that there exists a model $\mathcal{H}$ and a cell $q \in Q_l$ in this model for which $\mathcal{H}, q \models \varphi$. We can prove, analogous to the proof of Lemma 2.9, that for a formula φ one needs to look at cells of dimension at least $|\varphi|_{dc}$. A more coarse approximation is to say that one needs all the sets Q_n with $n \leq |\varphi| - |\varphi|_{uc}$. Thus, we can safely assume $l \leq |\varphi| - |\varphi|_{uc}$.

From Lemma 2.9 we know that we need to consider only the sets Q_m for $m \leq l + |\varphi|_{uc} \leq |\varphi|$, and all other sets Q are empty. From Lemma 2.8 we know that we can build a filtration model $\mathcal{H}^f$ s.t. the formula φ is still satisfiable and, by Lemma 2.7, we know that all the sets Q_m^f have a finite number of cells. Thus we can sum up all the cells in all the Q_m^f, with $m \leq |\varphi|$.

Corollary 2.11 (decidability). *Deciding the satisfiability of a HDML formula φ is done in space at most $\sum_{n=0}^{|\varphi|} 2^{|\varphi|\cdot N}$ where N is defined in Theorem 2.10.*

In the following we go on giving an axiomatic system for *HDML* and prove it sound and complete. In Table 2 we give a set of axioms and rules of inference for *HDML*. If a formula is *derivable* in this axiomatic system we write $\vdash \varphi$. We say that a formula φ is derivable from a set of formulas S iff $\vdash \psi_1 \wedge \cdots \wedge \psi_n \rightarrow \varphi$ for some $\psi_1, \ldots, \psi_n \in S$ (we write equivalently $S \vdash \varphi$). A set of formulas S is said to be *consistent* if $S \not\vdash \bot$, otherwise it is said to be *inconsistent*. A consistent set S is called *maximal* iff all sets S', with $S \subset S'$, are inconsistent.

Theorem 2.12 (soundness). *The axiomatic system of Table 2 is sound. Formally $\forall\varphi : \vdash \varphi \Rightarrow \models \varphi$.*

Proof (sketch). The proof tests that all axioms are valid and that all inference rules preserve validity. We check only the non-standard axioms (A5) to (A9').

We fix now some terminology and notation. Denote by $\neg\mathcal{C}(\varphi) = \mathcal{C}(\varphi) \cup \{\neg\varphi' \mid \varphi' \in \mathcal{C}(\varphi)\}$ the set of subformulas, as in Definition 2.4, together with their negated forms. A set of formulas A is called an *atom* for φ if A is a maximal consistent subset of $\neg\mathcal{C}(\varphi)$. Denote $At(\varphi)$ the set of all atoms for φ. Atoms are sets of formulas which are free of immediate propositional inconsistencies (like $\phi \wedge \neg\phi$).

Axiom schemes:

(A1) All instances of propositional tautologies.

(A2) $\{\}\bot \leftrightarrow \bot$	(A2') $\langle\rangle\bot \leftrightarrow \bot$
(A3) $\{\}(\varphi \vee \varphi') \leftrightarrow \{\}\varphi \vee \{\}\varphi'$	(A3') $\langle\rangle(\varphi \vee \varphi') \leftrightarrow \langle\rangle\varphi \vee \langle\rangle\varphi'$
(A4) $\{\![\,]\!\}\varphi \leftrightarrow \neg\{\}\neg\varphi$	(A4') $[\,]\varphi \leftrightarrow \neg\langle\rangle\neg\varphi$
(A5) $\langle\rangle i \rightarrow \langle\rangle^i\top \;\; \forall i \in \mathbb{N}^*$	(A5') $\{\}^i\top \rightarrow \{\}i \;\; \forall i \in \mathbb{N}^*$
(A6) $\langle\rangle[\,]\varphi \rightarrow [\,]\langle\rangle\varphi$	(A6') $\{\}\{\![\,]\!\}\varphi \rightarrow \{\![\,]\!\}\{\}\varphi$
(A7) $\{\}[\,]\varphi \rightarrow [\,]\{\}\varphi$	(A7') $\langle\rangle\{\![\,]\!\}\varphi \rightarrow \{\![\,]\!\}\langle\rangle\varphi$
(A8) $\{\}\langle\rangle^i\top \rightarrow \{\![\,]\!\}\langle\rangle^i\top \;\; \forall i \in \mathbb{N}$	(A8') $\langle\rangle\langle\rangle^i\top \rightarrow [\,]\langle\rangle^i\top \;\; \forall i \in \mathbb{N}$
(A9) $\langle\rangle^i\top \rightarrow \{\![\,]\!\}\langle\rangle\langle\rangle^i\top \;\; \forall i \in \mathbb{N}$	(A9') $\{\}\langle\rangle\langle\rangle^i\top \rightarrow \langle\rangle^i\top \;\; \forall i \in \mathbb{N}$

Inference rules:

$$\text{(R1)}\;\; \frac{\varphi \qquad \varphi \rightarrow \varphi'}{\varphi'}\;\text{(MP)} \qquad \text{(R2)}\;\; \frac{\varphi \rightarrow \varphi'}{\{\}\varphi \rightarrow \{\}\varphi'}\;\text{(D)} \qquad \text{(R2')}\;\; \frac{\varphi \rightarrow \varphi'}{\langle\rangle\varphi \rightarrow \langle\rangle\varphi'}\;\text{(D)}$$

(R3) Uniform variable substitution.

Table 2. Axiomatic system for $HDML$.

Lemma 2.13 (properties on atoms). *Standard results for atoms tell us that for some formula φ and any atom $A \in At(\varphi)$ then we have that:*

(i). for all $\psi \in \neg\mathcal{C}(\varphi)$ then only one of ψ or $\neg\psi$ are in A;

(ii). for all $\psi \rightarrow \psi' \in \neg\mathcal{C}(\varphi)$ then $\psi \rightarrow \psi' \in A$ iff whenever $\psi \in A$ then $\psi' \in A$;

(iii). if $\psi \in \neg\mathcal{C}(\varphi)$ and ψ is consistent then there exists an $A \in At(\varphi)$ s.t. $\psi \in A$; (This is an analog of Lindenbaum's Lemma.)

(iv). any consistent set of formulas $S \subseteq \neg\mathcal{C}(\varphi)$ can be grown to an atom $A_S \in At(\varphi)$.

Definition 2.14 (canonical saturated HDA). *A HDA is called* canonical *for the formula φ if a* canonical labeling *$\lambda : Q \rightarrow At(\varphi)$ can be attached to the HDA. A labeling is* canonical *if the following conditions hold:*

1. *for any $q_n \in Q_n, q_{n-1} \in Q_{n-1}$, for some $n > 0$, and $\forall 0 \leq i \leq n$, if $s_i(q_n) = q_{n-1}$ then $\forall\psi \in \neg\mathcal{C}(\varphi)$ if $\psi \in \lambda(q_n)$ then $\{\}\psi \in \lambda(q_{n-1})$,*
2. *for any $q_n \in Q_n, q_{n-1} \in Q_{n-1}$, for some $n > 0$, and $\forall 0 \leq i \leq n$, if $t_i(q_n) = q_{n-1}$ then $\forall\psi \in \neg\mathcal{C}(\varphi)$ if $\psi \in \lambda(q_{n-1})$ then $\langle\rangle\psi \in \lambda(q_n)$.*

A canonical HDA is called saturated *if:*

1. *whenever $\{\}\psi \in \lambda(q_{n-1})$ then $\exists q_n \in Q_n$ and $\exists 0 \leq i \leq n$ s.t. $s_i(q_n) = q_{n-1}$ and $\psi \in \lambda(q_n)$,*
2. *whenever $\langle\rangle\psi \in \lambda(q_n)$ then $\exists q_{n-1} \in Q_{n-1}$ and $\exists 0 \leq i \leq n$ s.t. $t_i(q_n) = q_{n-1}$ and $\psi \in \lambda(q_{n-1})$.*

Lemma 2.15 (truth lemma). *In a canonical saturated HDA $\mathcal{H}$ with the valuation defined as $\mathcal{V}(q_n) = \{\phi \in \Phi_B \mid \phi \in \lambda(q_n)\}$, it holds that $\mathcal{H}, q_n \models \psi$ iff $\psi \in \lambda(q_n)$.*

To prove completeness of the axiomatic system all that remains is to show that for any consistent formula φ we can build such a canonical saturated HDA. While building the canonical saturated HDA we constantly struggle to saturate the HDA (that we work with) while respecting the canonicity. Such not saturated $HDAs$ are called *defective*, as they may have defects, which we formally define below. But important is that any of these defects can be repaired. This is what the repair lemma does, using the two *enriching* and *lifting* constructions. The completeness theorem then shows that while starting with a minimal canonical HDA we can incrementally build a defect free canonical HDA.

Definition 2.16 (defects). *There are two types of defects for $\mathcal{H}$ (corresponding to a violation of a saturation condition):*

- *a* D1 defect *of $\mathcal{H}$ is a cell $q_n \in Q_n$ with $\{\}\psi \in \lambda(q_n)$ for which there is no $q_{n+1} \in Q_{n+1}$ and no $1 \leq i \leq n+1$, with $s_i(q_{n+1}) = q_n$ and $\psi \in \lambda(q_{n+1})$;*
- *a* D2 defect *of $\mathcal{H}$ is a cell $q_n \in Q_n$ with $\langle\rangle\psi \in \lambda(q_n)$ for which there is no $q_{n-1} \in Q_{n-1}$ and no $1 \leq i \leq n-1$, with $t_i(q_n) = q_{n-1}$ and $\psi \in \lambda(q_{n-1})$.*

For two $HDAs$, $\mathcal{H}_1$ and $\mathcal{H}_2$ we say that $\mathcal{H}_2$ *extends* $\mathcal{H}_1$ (written $\mathcal{H}_2 \rhd \mathcal{H}_1$) iff $\mathcal{H}_2$ has all the cells (with the same labels) and maps of $\mathcal{H}_1$ and possibly some new cells and maps (i.e., some extra structure).

Lemma 2.17 (enriching construction). *For a canonical model $\mathcal{H}$, there exists a construction (see Appendix), which we call* enriching of the $\mathcal{H}$ w.r.t. q and a formula $\{\}\varphi \in \lambda(q)$*, builds a model $\mathcal{H}'$ which is canonical and extends $\mathcal{H}$ (i.e., $\mathcal{H}' \rhd \mathcal{H}$).*

The *enriching construction* adds one new cell that has q as one of its sources and is labeled with an atom containing φ. Moreover, all the other maps of this new cell need to be added, together with all the necessary new cells, respecting the cubical laws.

The following *lifting construction* lifts all the cells of each level one level up by adding one new s and t map to each. The cubical laws make sure that these new maps reach only new cells; none of the old cells (that are lifted) are involved in these new instances of the cubical laws. We need to be careful how we label all these new cells s.t. the canonicity is respected for the extended $\mathcal{H}'$.

Lemma 2.18 (lifting construction). *For a canonical model $\mathcal{H}$, there exists a construction (see Appendix), which we call* lifting of the $\mathcal{H}$ w.r.t. q and a formula $\langle\rangle\varphi \in \lambda(q)$*, builds a model $\mathcal{H}'$ which is canonical and extends $\mathcal{H}$ (i.e., $\mathcal{H}' \rhd \mathcal{H}$).*

Lemma 2.19 (repair lemma). *For any canonical HDA $\mathcal{H}$ that has a defect we can build a corresponding $\mathcal{H}'$ which is canonical and does not have this defect.*

Proof. Consider that the canonical $\mathcal{H}$ from the statement has a defect of type D1. Apply the *enriching construction* to $\mathcal{H}$ w.r.t. the defective cell q_n and the formula ψ (where $\{\}\psi \in \lambda(q_n)$). The enriching lemma ensures that the new model $\mathcal{H}'$ extends $\mathcal{H}$ and is canonical. It is clear that the enriched model $\mathcal{H}'$ does not have the defect that $\mathcal{H}$ had.

Consider that the canonical $\mathcal{H}$ from the statement has a defect of type D2. Apply the *lifting construction* to $\mathcal{H}$ w.r.t. the defective cell q_n (for which $\langle\rangle\psi \in \lambda(q_n)$), to obtain, cf. lifting lemma, a canonical $\mathcal{H}'$ that extends $\mathcal{H}$. It is clear that the new model does not have the defect that $\mathcal{H}$ had.

Theorem 2.20 (completeness). *The axiomatic system of Table 2 is complete. Formally $\forall\varphi : \models \varphi \;\Rightarrow\; \vdash \varphi$.*

Proof. Using the truth lemma 2.15, the proof amounts to showing that for any consistent formula φ we can build a canonical saturated $\mathcal{H}_\varphi$ that has a cell labeled with an atom that contains φ. We construct $\mathcal{H}_\varphi$ in steps starting with $\mathcal{H}^0_\varphi$ which contains only one cell q^0_0 of dimension 0 labeled with an atom containing φ, i.e., $\lambda(q^0_0) = A_\varphi$. Trivially, $\mathcal{H}^0_\varphi$ is canonical. The cells used to construct our model are picked (in the right order) from the following sets $S_i = \{q^j_i \mid j \in \omega\}$ where $i \in \omega$ corresponds to the dimension i. Any of these cells may have defects and thus, we list all the defects, i.e., all the cells, and try to repair them in increasing order (i.e., we treat first defects on level 0 and continue upwards).

At some step $n \geq 0$ in the construction we consider $\mathcal{H}_\varphi^n = (Q^n, \overline{s^n}, \overline{t^n}, l^n)$ canonical. If $\mathcal{H}_\varphi^n$ is not saturated then pick the smallest defect cell of $\mathcal{H}_\varphi^n$. For a D1 defect, i.e., a cell $q_k \in Q_k$ and formula $\{\}\psi \in \lambda(q_k)$, apply enrich(k,$q_k$,$\psi$) and obtain a model $\mathcal{H}_\varphi^{n+1}$ which is canonical, cf. Lemma 2.17, and does not have the D1 defect, cf. Lemma 2.19. For a D2 defect apply the lifting construction to remove the defect. Moreover, any repaired defect will never appear in any extension model, independent of how many times we apply the enriching or lifting constructions. Both enriching and lifting pick their new cells from S in increasing order. We obtain $\mathcal{H}_\varphi$ as a limit construction from all the $\mathcal{H}_\varphi^n$; i.e., $\mathcal{H}_\varphi = (Q, \overline{s}, \overline{t}, l)$ as $Q = \bigcup_{n\in\omega} Q^n,\ \overline{s} = \bigcup_{n\in\omega} \overline{s^n},\ \overline{t} = \bigcup_{n\in\omega} \overline{t^n},\ l = \bigcup_{n\in\omega} l^n$.

3 Examples of Encodings into Higher Dimensional Modal Logic

This section serves to exemplify two main ways of using *HDML*. One usage is as a highly expressive logic in which many other logics for different concurrency models can be encoded; in this respect we study the relation of *HDML* with standard modal logic, LTL, and with linear time temporal logic over Mazurkiewicz traces LTrL. The other usage is as a general theoretical framework for studing different logics for different concurrency models (that can be expressed as some class of *HDA*) and their interrelation. This is done by finding the apropriate restrictions of *HDA* and *HDML* and investigating their relations in *HDML*.

3.1 Encoding standard modal logic into HDML

Lemma 3.1 (Kripke structures). *The class of Kripke structures is captured by the class of higher dimensional structures where all sets Q_n, for $n > 1$, are empty.*

Proof. Essentially this result is found in [21]. A *HDA* $K = (Q_0, Q_1, s_1, t_1, l)$ is a special case of *HDAs* where all $Q_n = \emptyset$ for $n > 1$. This is the class of *HDAs* that encode Kripke frames. Because Q_2 (and all other cells of higher dimension) is empty there are no cubical laws applicable. Therefore, there is no geometric structure on K. Moreover, the restriction on the labeling function l is not applicable (as Q_2 is empty). Add to such a *HDA* a valuation function $\mathcal{V}$ to obtain a Kripke model $(Q_0, Q_1, s_1, t_1, l, \mathcal{V})$.

Proposition 3.2 (axiomatization of Kripke *HDAs*). *The class of higher dimensional structures corresponding to Kripke structures (from Lemma 3.1) is axiomatized by:*

$$\models \neg\langle\rangle\top \rightarrow \{\!\!\{\}\!\!\}\{\!\!\{\}\!\!\}\bot$$

Proof (sketch). For any *HDA* $\mathcal{H}$ and any $q \in Q$ a cell of any dimension, we prove the double implication: $\mathcal{H}, q \models \neg\langle\rangle\top \rightarrow \{\!\!\{\}\!\!\}\{\!\!\{\}\!\!\}\bot$ iff $\mathcal{H}$ is as in Lemma 3.1.

Theorem 3.3 (standard modal logic). *Consider the syntactic definition $\Diamond\varphi \triangleq \{\}\langle\rangle\varphi$. The language of* standard modal logic *uses only $\Diamond$ and is interpreted only over higher dimensional structures as defined in Lemma 3.1 and only in cells of Q_0.*

Proof (sketch). First we check that we capture exactly the semantics of standard modal logic. Second we check that we recover the axiomatic system of standard modal logic for $\Diamond$ from the axiomatic system of *HDML*.

3.2 Adding an Until operator and encoding LTL

The basic temporal logic is the logic with only the *eventually* operator (and the dual *always*). This language is expressible in the standard modal logic [3] and thus is expressible in $HDML$ too. It is known that the *Until* operator $\mathcal{U}$ adds expressivity to LTL (*eventually* and *always* operators can be encoded with $\mathcal{U}$ but not the other way around).

The *Until* operator cannot be encoded in $HDML$ because of the local behavior of the during and terminate modalities; similar arguments as in modal logic about expressing $\mathcal{U}$ apply to $HDML$ too. The *Until* modality talks about the whole model (about all the configurations of the system) in an existential manner. More precisely, the *Until* says that there must exist some configuration in the model, reachable from the configuration where *Until* is evaluated, satisfying some property φ and in all the configurations on all the paths reaching the φ configuration some other property ψ must hold. Hence we need a notion of *path* in a HDA.

Definition 3.4 (paths in $HDAs$). *A* simple step *in a HDA is either $q_{n-1} \xrightarrow{s_i} q_n$ with $s_i(q_n) = q_{n-1}$ or $q_n \xrightarrow{t_i} q_{n-1}$ with $t_i(q_n) = q_{n-1}$, where $q_n \in Q_n$ and $q_{n-1} \in Q_{n-1}$ and $1 \leq i \leq n$. A* path *$\pi \triangleq q^0 \xrightarrow{\alpha^0} q^1 \xrightarrow{\alpha^1} q^2 \xrightarrow{\alpha^2} \dots$ is a sequence of single steps $q^j \xrightarrow{\alpha^j} q^{j+1}$, with $\alpha^j \in \{s_i, t_i\}$. We say that $q \in \pi$ iff $q = q^j$ appears in one of the steps in π. The first cell in a path is denoted $st(\pi)$ and the ending cell in a finite path is $en(\pi)$.*

In the same spirit as done for temporal logic we boost the expressivity of $HDML$ by defining an $\mathcal{U}$ operator over higher dimensional structures. Using this operator we can encode the standard *Until* operator of LTL.

Definition 3.5 (Until operator). *Define an Until operator $\varphi \mathcal{U} \varphi'$ which is interpreted over a HDA in a cell as below:*

$$\mathcal{H}, q \models \varphi \mathcal{U} \varphi' \text{ iff } \exists \pi \in \mathcal{H} \text{ s.t. } st(\pi) = q \wedge en(\pi) = q', \\ \mathcal{H}, q' \models \varphi', \text{ and } \forall q'' \in \pi, q'' \neq q' \text{ then } \mathcal{H}, q'' \models \varphi.$$

Proposition 3.6 (modeling LTL). *The LTL Until modality is encoded syntactically by $\varphi \overline{\mathcal{U}} \varphi' \triangleq (\varphi \vee \langle\rangle \top) \mathcal{U} (\varphi' \wedge \neg \langle\rangle \top)$ when $\overline{\mathcal{U}}$ is interpreted only in states of Kripke $HDAs$ as in Lemma 3.1.*

3.3 Partial order models and their logics in HDML

This section is mainly concerned with Mazurkiewicz traces [8] as a model of concurrency based on partial orders, because of the wealth of logics that have been developed for it [9,20]. Higher dimensional automata are more expressive than most of the partial orders models (like Mazurkiewicz traces, pomsets [13], or event structures [10]) as studied in [16, 21]. The works of [16, 17, 21] show (similar in nature) how event structures can be encoded in higher dimensional automata. Mazurkiewicz traces are a particular class of event structures, precisely defined in [19]. We use this presentation, as a restricted partial order, of Mazurkiewicz traces.

In the following we give definitions and standard results on partial orders, event structures, and Mazurkiewicz traces which are needed for the development of the higher dimensional modal logic for these models, in particular for Mazurkiewicz traces. In few words, we isolate the class of higher dimensional automata corresponding to Mazurkiewicz traces (and to partial orders or event structures in general) as the models of the $HDML$. Then we restrict $HDML$ to get exactly the logics over Mazurkiewicz traces (we focus on the logics presented in [4, 20]) and over general partial orders (like ISTL of [1]).

Definition 3.7 (partial orders). *A partially ordered set (or* poset*) is a set E equipped with a partial order $\leq$, $(E, \leq)$. The* history *of an element $e \in E$ (denoted $\downarrow e$) is $\downarrow e = \{e' \mid e' \leq e\}$. The notion of history is extended naturally to a set of elements $C \subseteq E$ (denoted $\downarrow C$). A* configuration *is a finite and history closed set of elements (i.e., $C = \downarrow C$). Denote by $\mathcal{C}$ the set of all configurations. The* immediate successor *relation $\lessdot \subseteq E \times E$ is defined as $e \lessdot e'$ iff $e \neq e'$ and $e \leq e'$ and $\forall e'' \in E$, $e \leq e'' \leq e'$ implies $e = e''$ or $e' = e''$. A Σ-*labeled poset *$(E, \leq, \lambda)$ is a poset with a labeling function $\lambda : E \to \Sigma$ which maps each element to a label from Σ. Define a* transition relation *on the configurations of a labeled poset as $\longrightarrow \subseteq \mathcal{C} \times \Sigma \times \mathcal{C}$ given by $C \xrightarrow{a} C'$ iff $\exists e \in E$ s.t. $\lambda(e) = a$ and $e \notin C$ and $C' = C \cup \{e\}$.*

When one sees the elements of E as the *events* of a system, the labels can be seen as the names of the actions that the events are instances of.

Definition 3.8 (Mazurkiewicz traces). *Consider a symmetric irreflexive* independence relation *$I \subseteq \Sigma \times \Sigma$ and its complement $D = \Sigma \times \Sigma \setminus I$, called the* dependence relation*. Mazurkiewicz traces are labeled posets restricted by the independence relation as follows:*

$\forall e \in E, \quad \downarrow e$ *is finite,*
$\forall e, e' \in E,\ e \lessdot e' \Rightarrow (\lambda(e), \lambda(e')) \in D,$
$\forall e, e' \in E,\ (\lambda(e), \lambda(e')) \in D \Rightarrow e \leq e'$ *or* $e' \leq e$.

Definition 3.9 (event structures). *Consider a symmetric irreflexive relation $\# \subseteq E \times E$. This* conflict relation *is added to a poset to form an* event structure *$(E, \leq, \#)$ where the following restrictions apply:*

$\forall e, e', e'' \in E,\ e \# e'$ *and* $e' \leq e''$ *implies* $e \# e''$,
$\forall e, e' \in E, \quad e \in C$ *and* $e \# e'$ *implies* $e' \notin C$.

An event structure is called finitary *iff $\forall e \in E$, $\downarrow e$ is finite.*

The second constraint on event structures says that the configurations of an event structure are conflict-free. Define the relation of concurrency for an event structure to be:

$$co = E \times E \setminus (\# \cup \leq \cup \leq^{-1}).$$

Proposition 3.10 (families of configurations). *A finitary event structure $(E, \leq, \#)$ is uniquely determined by its family of configurations $\mathcal{C}_E$ (denoted $(E, \mathcal{C}_E)$).*

Proof (sketch). This result is found in [17]. We summarize here the results leading to it.

The two relations $e \leq e'$ and $e \# e'$ are mutually exclusive, because, otherwise, the set $\downarrow e'$ would not be a configuration (because of the second constraint of Definition 3.9).

If two events e, e' do not appear together in any configuration of $\mathcal{C}_E$ then $e \# e'$ ($e \# e'$ iff $\nexists C \in \mathcal{C}_E$ s.t. $e, e' \in C$).

If in any configuration where e' exists, e exists too then $e \leq e'$ ($e \leq e'$ iff $\forall C \in \mathcal{C}_E, e' \in C \Rightarrow e \in C$).

We usually use a labeled poset and work with labeled event structures $(E, \leq, \#, \lambda)$, or $(E, \mathcal{C}_E, \lambda)$ when using their corresponding family of configurations.

Proposition 3.11 (traces as event structures). *Any Mazurkiewicz trace, as in Definition 3.8, corresponds to a* trace configuration structure*, which is a labeled event structure $(E, \mathcal{C}_E, \lambda)$ that respects the following restriction:*

λ *is* a nice labeling *and* context-independent,

where nice labeling means
$\forall e, e' \in E,\ \lambda(e) = \lambda(e') \Rightarrow e \leq e'$ *or* $e' \leq e$
and context-independent means
$\forall a, b \in \Sigma,\ (\lambda^{-1}(a) \times \lambda^{-1}(b)) \cap co \neq \emptyset \ \Rightarrow\ (\lambda^{-1}(a) \times \lambda^{-1}(b)) \cap \lessdot = \emptyset\,.$

Proof (sketch). This result is essentially found in [9,19]. We remind how one gets the independence relation of a Mazurkiewicz trace from a trace configuration structure:
$I = \{(a,b) \mid (\lambda^{-1}(a) \times \lambda^{-1}(b)) \cap co \neq \emptyset\}$.

One can view a configuration as a valuation of events $E \to \{0,1\}$, and thus we can view an event structure as a valuation $f_E : 2^E \to \{0,1\}$, which selects only those configurations that make the event structure.

The terminology that we adopt now steams from the Chu spaces representation of $HDAs$ [16,17]. We fix a set E, which for our purposes denotes events. Consider the class of $HDAs$ which have a single hypercube of dimension $|E|$, hence each event represents one dimension in the HDA. This hypercube is denoted 3^E, in relation to 2^E, because in the HDA case each event may be in three phases, *not started*, *executing*, and *terminated* (as opposed to only terminated or not started). The valuation from before becomes now $E \to \{0, \frac{1}{2}, 1\}$, where $\frac{1}{2}$ means executing. The set of three values is linearly ordered $0 < \frac{1}{2} < 1$ to obtain an *acyclic* HDA [17], and all cells of 3^E (i.e., the configurations) are ordered by the natural lifting of this order pointwise. The dimension of a cell is equal to the number of $\frac{1}{2}$ in its corresponding valuation.

Notation: In the context of a single hypercube 3^E we denote the cells of the cube by lists of $|E|$ elements $e_1 e_2 \dots e_{|E|}$ where each e_i takes values in $\{0, \frac{1}{2}, 1\}$ and represents the status of the i^{th} event of the HDA.

With the above conventions, the cells of dimension 0 (i.e., the states of the HDA) are denoted by the corresponding valuation restricted to only the two values $\{0,1\}$; and correspond to the configurations of an event structure. The set of states of such a HDA is partially ordered by the order $<$ we defined before. In this way, from the hypercube 3^E we can obtain any family of configurations $\mathcal{C}_E$ by removing all 0-dimensional cells that represent a configuration $C \notin \mathcal{C}_E$.[2] By Proposition 3.10 we can reconstruct the event structure.

In Definition 2.3 the interpretation of the during and terminate modalities of $HDML$ did not take into consideration the labeling of the HDA. The labeling was used only for defining the geometry of concurrency of the HDA. Now we make use of this labeling function in the semantics of the labeled modalities of Definition 3.13. But first we show how the labeling extends to cells of any dimension.

Definition 3.12 (general labeling). *Because of the condition* $l(s_i(q)) = l(t_i(q))$ *for all* $q \in Q_2$*, all the edges* $e_1 \dots e_{i-1} \frac{1}{2} e_{i+1} \dots e_{|E|}$*, with* $e_j \in \{0,1\}$ *for* $j \neq i$*, have the same label. Denote this as the label* l_i*. The label of a general cell* $q \in Q_n$ *is the multiset of* n *labels* $l_{j_1} \dots l_{j_n}$ *where the* j*'s are exactly those indexes in the representation of* q *for which* e_j *has value* $\frac{1}{2}$*.*

As is the case with multi-modal logics or propositional dynamic logics, we extend $HDML$ to have a multitude of modalities indexed by some alphabet Σ (the alphabet of the HDA in our case). This will be the same alphabet as that of the Mazurkiewicz trace represented by the HDA.

[2] We remove also all those cells of higher dimension that are connected with the 0-dimensional cells that we have removed.

Definition 3.13 (labeled modalities). *Consider two labeled modalities* during $\{a\}\varphi$ *and* terminate $\langle a\rangle\varphi$ *where* $a \in \Sigma$ *is a label from a fixed alphabet. The interpretation of the labeled modalities is given below:*

$\mathcal{H}, q \models \{a\}\varphi$ *iff assuming* $q \in Q_n$ *for some* n, $\exists q' \in Q_{n+1}$ *s.t.* $s_i(q') = q$ *for some* $1 \leq i \leq n$, $l(q') = l(q)a$ *and* $\mathcal{H}, q' \models \varphi$.

$\mathcal{H}, q \models \langle a\rangle\varphi$ *iff assuming* $q \in Q_n$ *for some* n, $\exists q' \in Q_{n-1}$ *s.t.* $t_i(q) = q'$ *for some* $1 \leq i \leq n$, $l(q) = l(q')a$ *and* $\mathcal{H}, q' \models \varphi$.

Having the labeled modalities one can get the unlabeled variants as a disjunction over all labels $\{\}\varphi \overset{\Delta}{=} \bigvee_{a\in\Sigma}\{a\}\varphi$.

In the remaining of this section we show how the LTrL logic of [20] is captured in the higher dimensional framework. This logic, as well as those presented in [4, 9], are interpreted in some particular configuration of a Mazurkiewicz trace (or of a partial order). We take the view of Mazurkiewicz traces as restricted labeled posets from Proposition 3.8 but we use their representation using their corresponding family of configurations as in Proposition 3.11. Therefore, we now interpret $HDML$ over restricted $HDAs$ as we discussed above.

Proposition 3.14 (encoding LTrL). *The language of LTrL consists of the propositional part of* $HDML$ *together with the syntactic definitions of the Until operator* $\overline{\mathcal{U}}$ *from Definition 3.6 and* $\overline{\langle a\rangle}\varphi \overset{\Delta}{=} \{a\}\langle a\rangle\varphi$ *for* $a \in \Sigma$*. When interpreted only in the states of a* HDA *representing a Mazurkiewicz trace this language has the same behavior as the one presented in [20]*

Proof. The states of the HDA are the configurations of the Mazurkiewicz trace. Thus, our definition of the LTrL language is interpreted in one trace at one particular configuration; as is done in [20]. The original semantics of LTrL uses transitions from one configuration to another labeled by an element from the alphabet Σ of the trace. It is easy to see that our syntactic definition of $\overline{\langle a\rangle}\varphi$ has the same interpretation as the one in [20]. The proof is similar to the proof of Theorem 3.3. The *Until* operator of [20] has the same definition as the one in standard LTL and thus we use the one defined in Proposition 3.6; the proof is easily adapted to the Mazurkiewicz traces setting.

4 Conclusion

We have introduced a modal logic called $HDML$ which is interpreted over higher dimensional automata. According to our knowledge, this has not been done before. The language of $HDML$ is simple, capturing both the notions of "during" and "after". The associated semantics is intuitive, accounting for the special geometry of the $HDAs$. An adaptation of the filtration method was needed to prove decidability. We have associated to $HDML$ an axiomatic system which incorporates the standard modal axioms and has extra only few natural axioms related to the cubical laws and to the dimensions of $HDAs$. This system was proven to be complete for $HDAs$.

We isolated axiomatically the class of $HDAs$ that encode Kripke structures and shown how standard modal logic is encoded into $HDML$ when interpreted only over these restricted $HDAs$. We then extended the expressiveness of $HDML$ by defining an *Until* operator over $HDAs$. Using this *Until* operator, the LTL was encoded into $HDML$ when interpreted over the Kripke $HDAs$.

As future work we are investigating a tableaux system for $HDML$. We are also trying to understand better the relation of $HDML$ with other logics for weaker models of concurrency like with the modal logic of [6] for event structures or other logics for Mazurkiewicz traces. Particularly interesting is how our results relate to the undecidability results of [2] or to the logic of [1].

Regarding the expressiveness of $HDML$, our current work focuses on finding the kind of bisimulation that is characterized by $HDML$ (and its extensions from Section 3). Standard bisimulations for concurrent systems like ST-bisimulation or split-bisimulation are not characterized by $HDML$ because of the *during modality* and the "during" behaviour of $HDML$.

Acknowledgements: I thank Martin Steffen and Olaf Owe for their useful comments.

References

1. R. Alur, K. L. McMillan, and D. Peled. Deciding Global Partial-Order Properties. *Formal Methods in System Design*, 26(1):7–25, 2005.
2. R. Alur and D. Peled. Undecidability of partial order logics. *Inf. Process. Lett.*, 69(3):137–143, 1999.
3. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theor. Comput. Sci*. Cambridge Univ. Press, 2001.
4. V. Diekert and P. Gastin. LTL Is Expressively Complete for Mazurkiewicz Traces. In *ICALP'00*, volume 1853 of *LNCS*, pages 211–222. Springer, 2000.
5. V. Diekert and P. Gastin. From local to global temporal logics over Mazurkiewicz traces. *Theor. Comput. Sci.*, 356(1-2):126–135, 2006.
6. K. Lodaya, M. Mukund, R. Ramanujam, and P. S. Thiagarajan. Models and Logics for True Concurrency. Technical Report IMSc-90-12, Inst. Mathematical Science, Madras, India, 1990.
7. K. Lodaya, R. Parikh, R. Ramanujam, and P. S. Thiagarajan. A Logical Study of Distributed Transition Systems. *Inf. Comput.*, 119(1):91–118, 1995.
8. A. W. Mazurkiewicz. Basic notions of trace theory. In *REX Workshop*, volume 354 of *LNCS*, pages 285–363. Springer, 1988.
9. M. Mukund and P. S. Thiagarajan. Linear Time Temporal Logics over Mazurkiewicz Traces. In *MFCS'96*, volume 1113 of *LNCS*, pages 62–92. Springer, 1996.
10. M. Nielsen, G. D. Plotkin, and G. Winskel. Petri nets, event structures and domains. In *Semantics of Concurrent Computation*, volume 70 of *LNCS*, pages 266–284. Springer, 1979.
11. A. Pnueli. The temporal logic of programs. In *FOCS'77*, pages 46–57. IEEE Computer Society Press, 1977.
12. V. R. Pratt. A Practical Decision Method for Propositional Dynamic Logic: Preliminary Report. In *STOC'78*, pages 326–337. ACM Press, 1978.
13. V. R. Pratt. Modeling Concurrency with Partial Orders. *J. Parallel Programming*, 15(1):33–71, 1986.
14. V. R. Pratt. Modeling concurrency with geometry. In *POPL'91*, pages 311–322, 1991.
15. V. R. Pratt. Chu spaces and their interpretation as concurrent objects. In *Computer Science Today: Recent Trends and Developments*, volume 1000 of *Lecture Notes in Computer Science*, pages 392–405. Springer, 1995.
16. V. R. Pratt. Higher dimensional automata revisited. *Math. Struct. Comput. Sci.*, 10(4):525–548, 2000.
17. V. R. Pratt. Transition and Cancellation in Concurrency and Branching Time. *Math. Struct. Comput. Sci.*, 13(4):485–529, 2003.
18. C. Prisacariu. Modal Logic over Higher Dimensional Automata – technicalities. Technical Report 393, Univ. Oslo, January 2010.
19. B. Rozoy and P. S. Thiagarajan. Event structures and trace monoids. *Theor. Comput. Sci.*, 91(2):285–313, 1991.
20. P. S. Thiagarajan and I. Walukiewicz. An Expressively Complete Linear Time Temporal Logic for Mazurkiewicz Traces. *Information and Computation*, 179(2):230–249, 2002.
21. R. J. van Glabbeek. On the Expressiveness of Higher Dimensional Automata. *Theor. Comput. Sci.*, 356(3):265–290, 2006.

A Proofs

Proposition A.1 (size of the closure). *The size of the subformula closure of a formula φ is linear in the size of the formula; i.e., $|\mathcal{C}(\varphi)| \leq |\varphi|$.*

Proof. The proof is easy, using structural induction and observing that for the atomic formulas the size of the closure is exactly 1, the size of the formula. For a compound formula like $\{\}\varphi$ the induction hypothesis says that $|\mathcal{C}(\varphi)| \leq |\varphi|$ which means $1 + |\mathcal{C}(\varphi)| \leq 1 + |\varphi|$.

Lemma A.2. *Any two sets $[p], [q] \in Q^f_n$, for some $n \in \mathbb{N}$, are disjoint.*

Proof. By induction on n.

The base case for $n = 0$ is easy as the definition of Q^f_0 results in the equivalence classes on Q_0 generated by the equivalence relation $\equiv$, which are disjoint.

Inductive step: Consider $[p], [q] \in Q^f_n$, for which we assume that $\exists r \in Q_n$ with $r \in [p]$ and $r \in [q]$. From the definition we get (1) $q \equiv r \equiv p$ and, (2) for any $1 \leq i \leq n$ and some fixed $[p'_i], [q'_i] \in Q^f_{n-1}$, $t_i(r) \in [p'_i]$ and $t_i(r) \in [q'_i]$. By the induction hypothesis we know that $[p'_i]$ and $[q'_i]$ are disjoint, which, together with (2) before, implies that $[p'_i] = [q'_i]$ for all $1 \leq i \leq n$. Because of this and (1) it implies that $[q] = [p]$. Therefore we have proven that if two sets $[p], [q] \in Q^f_n$ have a cell in common then they must be the same. (Note that a similar treatment of s_i is needed.)

Lemma A.3.

1. *The definitions of s^f_i and t^f_i are that of* maps *(as required in a higher dimensional structure).*
2. *The s^f_i and t^f_i respect the* cubical laws *of a higher dimensional structure.*

Proof. For 1. we give the proof only for t^f_i, as the proof for s^f_i is analogous. We use *reductio ad absurdum* and assume, for some $[q] \in Q^f_n$, that $t^f_i([q]) = [p]$ and $t^f_i([q]) = [p']$ with $[p] \neq [p']$ and $[p], [p'] \in Q^f_{n-1}$. From the definition we have that for all $q \in [q]$ both $t_i(q) \in [p]$ and $t_i(q) \in [p']$. From Lemma A.2 we know that $[p]$ and $[p']$ are disjoint and we know that t_i is a map (i.e., the outcome is unique), therefore we have the contradiction.

For 2. we have to prove, for some arbitrary chosen $[q] \in Q^f_n$ and for any $1 \leq i < j \leq n$ that

$$t^f_i(t^f_j([q])) = t^f_{j-1}(t^f_i([q])).$$

(Note that t^f_i on the left side is different than the t^f_i on the right side, as the left one is applied to elements of Q^f_{n-1} whereas the right one is applied to elements of Q^f_n.) The other three kinds of cubical laws are treated analogous only that one needs to reason with the s_i maps too.

Assume, wlog. because the opposite assumption would follow analogous reasoning, that $t^f_i(t^f_j([q])) = [q_{n-2}]$ with $[q_{n-2}] \in Q^f_{n-2}$. This leads to considering that $t^f_j([q]) = [q_{n-1}]$ with $[q_{n-1}] \in Q^f_{n-1}$, and $t^f_i([q_{n-1}]) = [q_{n-2}]$. From the definition we have both:
(1) $\forall q \in [q] : t_j(q) \in [q_{n-1}]$,
(2) $\forall q \in [q_{n-1}] : t_i(q) \in [q_{n-2}]$.
Therefore, from the two we have that
(3) $\forall q \in [q] : t_i(t_j(q)) \in [q_{n-2}]$.

We want to prove that $[q_{n-2}] = t^f_{j-1}(t^f_i([q]))$, for which we can assume, wlog., that $t^f_i([q]) = [q'_{n-1}]$ for some $[q'_{n-1}] \in Q^f_{n-1}$. Therefore, it amounts to proving that $t^f_{j-1}([q'_{n-1}]) = [q_{n-2}]$. For this it is enough to find some $p \in [q'_{n-1}]$ s.t. $t_{j-1}(p) \in [q_{n-2}]$, because by the Definition 2.5 (of the t_i maps) it means that $\forall p \in [q'_{n-1}]$ it holds that $t_{j-1}(p) \in [q_{n-2}]$, i.e., our desired result.

From the assumption we have that $\forall q \in [q] : t_i(q) \in [q'_{n-1}]$. Pick one of these $t_i(q)$ and claim this to be the $p \in [q'_{n-1}]$ we are looking for. From the cubical laws for the initial $\mathcal{H}$ model we know that for any $q \in [q]$, $t_i(t_j(q)) = t_{j-1}(t_i(q)) = t_{j-1}(p)$. Because of (3) we have that $t_{j-1}(p) \in [q_{n-2}]$, and thus our claim is proven; i.e, t_{j-1} applied to the element $t_i(q)$ that we picked from $[q'_{n-1}]$, is in $[q_{n-2}]$.

Proof (of Lemma 2.8). By induction on the structure of the formula ψ.

Base case: For $\psi = \phi$ is immediate from the definition of $\mathcal{V}^f$.

Inductive step: The case for $\rightarrow$ is straightforward making use of the induction hypothesis because the set $\mathcal{C}(\varphi)$ is closed under subformulas.

Take now $\psi = \langle\rangle\psi'$ and we prove that $\mathcal{H}, q \models \langle\rangle\psi'$ iff $\mathcal{H}^f, [q] \models \langle\rangle\psi'$. Considering the *only if* implication we assume that (cf. definition of satisfiability from Table 1) $\exists q' \in Q_{n-1} : t_i(q) = q' \wedge q' \models \psi'$ for some $1 \leq i \leq n$, and have to prove that $\exists [p] \in Q^f_{n-1} : t^f_i([q]) = [p] \wedge [p] \models \psi'$ for some $1 \leq i \leq n$. Because $q \in [q]$ and $t_i(q) = q'$, using the definition of $[q]$ it implies that for all $q \in [q]$ is that $t_i(q) \in [q']$ which, by the definition of t^f_i, implies that $t^f_i([q]) = [q']$. (Thus we have found the $[q'] \in Q^f_{n-1}$.) From the induction hypothesis we have that $\mathcal{H}, q' \models \psi'$ implies that $\mathcal{H}^f, [q'] \models \psi'$. This ends the proof.

Consider now the *if* implication and assume $\exists [p] \in Q^f_{n-1} : t^f_i([q]) = [p] \wedge [p] \models \psi'$ for some $1 \leq i \leq n$. From the definition of t^f_i we have that $t_i(q) \in [p]$; which is the same as picking some $p' \in [p]$ with $t_i(q) = p'$. From the induction hypothesis we know that $\mathcal{H}^f, [p] \models \psi'$ iff $\mathcal{H}, p \models \psi'$ for any $p \in [p]$ (in particular $\mathcal{H}, p' \models \psi'$). Thus $\exists p' \in Q_{n-1} : t_i(q) = p' \wedge \mathcal{H}, p' \models \psi'$ for some $1 \leq i \leq n$, finishing the proof.

When we take $\psi = \{\}\psi'$ we use analogous arguments as in the proof of $\langle\rangle\psi'$. In this case we work with the definition of s^f_i and we look for cells of higher dimension (instead of lower dimension).

Proof (of Lemma 2.9). By induction on the structure of the formula φ.

Base case: For $\phi \in \Phi_B$ and $\perp$ the evaluation is in the same cell q and thus all the cells of dimension higher than k are not important and can be empty.

Inductive step: For $\varphi_1 \rightarrow \varphi_2$ the semantics says that whenever $\mathcal{H}, q \models \varphi_1$ then $\mathcal{H}, q \models \varphi_2$. From the induction hypothesis we have that all cells of dimension greater that $k + |\varphi_1|_{uc}$ (respectively $k + |\varphi_2|_{uc}$) are not important for checking φ_1 (respectively φ_2). Thus it is a safe approximation to consider all the cells of at most dimension $max(k + |\varphi_1|_{uc}, k + |\varphi_2|_{uc}) = k + |\varphi_1 \rightarrow \varphi_2|_{uc}$ and all sets Q_m of greater dimension can be empty.

For $\{\}\varphi$ the semantics says that we need to check the formula φ in cells of dimension one greater, i.e., $q_{k+1} \models \varphi$. From the induction hypothesis we know that for checking $q_{k+1} \models \varphi$ it is enough to have only cells of most dimension $k + 1 + |\varphi|_{uc} = k + |\{\}\varphi|_{uc}$ (where all other cells can be removed).

For $\langle\rangle\varphi$ the semantics says that we need to check $q_{k-1} \models \varphi$, that is, in cells of immediately lower dimension. By the induction hypothesis this means that it is enough to consider only cells of at most dimension $k - 1 + |\varphi|_{uc}$ for which $k + |\varphi|_{uc}$ is a safe approximation.

Proposition A.4 (theorems). *The following are derivable in the axiomatic system of Table 2:*

$\vdash \{\}(\varphi \to \varphi') \to (\{\}\varphi \to \{\}\varphi')$,
$\vdash \langle\rangle(\varphi \to \varphi') \to (\langle\rangle\varphi \to \langle\rangle\varphi')$,
$\vdash (\langle\rangle[\,]\varphi \wedge \langle\rangle[\,]\neg\varphi) \to \bot$,
$\vdash \langle\rangle[\!\{\}\!]\bot \to [\!\{\}\!]\bot$,
$\vdash \{\}\top \to [\,]\{\}\top$,
$\vdash \langle\rangle\top \to (\{\}[\,]\varphi \to \langle\rangle\{\}\varphi)$,
$\vdash \{\}\top \to (\langle\rangle[\!\{\}\!]\varphi \to \{\}\langle\rangle\varphi)$,
$\vdash [\!\{\}\!]\langle\rangle\top$.

Moreover, one can use the following derived rules:

$$\frac{\varphi}{[\!\{\}\!]\varphi}\,, \qquad \frac{\varphi}{[\,]\varphi}\,,$$

$$\frac{\varphi \to \varphi'}{[\!\{\}\!]\varphi \to [\!\{\}\!]\varphi'}\,, \qquad \frac{\varphi \to \varphi'}{[\,]\varphi \to [\,]\varphi'}\,.$$

Proof. The first two theorems are derivable as in standard modal logic only using the standard axioms (A2)-(A3'). The derived rules are also as in standard modal logic. The third theorem is a consequence of (A6): $\langle\rangle[\,]\varphi \wedge \langle\rangle[\,]\neg\varphi \overset{(A6)}{\to} [\,]\langle\rangle\varphi \wedge \langle\rangle[\,]\neg\varphi \overset{SML}{\to} \langle\rangle(\langle\rangle\varphi \wedge [\,]\neg\varphi) \overset{SML}{\to} \langle\rangle\langle\rangle(\varphi \wedge \neg\varphi) \overset{(A2')}{\to} \bot$. The forth theorem is a consequence of (A7'): $\langle\rangle[\!\{\}\!]\bot \overset{(A7')}{\to} [\!\{\}\!]\langle\rangle\bot \overset{(A2')}{\to} [\!\{\}\!]\bot$. The fifth theorem is a consequence of the theorem before by contraposition. The sixth theorem is a consequence of (A7): $(\{\}[\,]\varphi \overset{(A7)}{\to} [\,]\{\}\varphi) \leftrightarrow (\{\}[\,]\varphi \wedge \langle\rangle\top \to [\,]\{\}\varphi \wedge \langle\rangle\top)$ but $[\,]\{\}\varphi \wedge \langle\rangle\top \to \langle\rangle\{\}\varphi$ which means that $(\{\}[\,]\varphi \wedge \langle\rangle\top \to \langle\rangle\{\}\varphi) \leftrightarrow \langle\rangle\top \to (\{\}[\,]\varphi \to \langle\rangle\{\}\varphi)$. The seventh theorem is derivable in an analogous way as the one above only that we use axiom (A7'). The eighth theorem is just the instantiation of axiom (A9) when $i = 0$ (i.e., $\langle\rangle^0\top \triangleq \top$).

Proof (of Theorem 2.12). Before proving soundness we should have some intuition about the non-standard axioms (A5) to (A9'). First consider the four axioms (A6) to (A7') that each relates to one of the four cubical laws. Axiom (A6) embodies the cubical law $t_i(t_j(q)) = t_{j-1}(t_i(q))$ (i.e., the cubical law where α is instantiated to t and β to t). Axiom (A6') embodies the cubical law where α and β are both instantiated to s (e.g., $s_i(s_j(q)) = s_{j-1}(s_i(q))$). The two axioms (A7) and (A7') relate to the cubical laws where α and β are instantiated differently, one to s and the other to t; e.g., $s_i(t_j(q)) = t_{j-1}(s_i(q))$.

The other axioms talk about the dimensions of the cells and about the division of the cells into layers Q_n.

– Axiom (A5) $\langle\rangle i \to \langle\rangle^i\top$ says that if in a cell there can be terminated at least i different events then this means that this cell has dimension at least i (i.e., one can go i levels down by $\langle\rangle^i\top$). This is natural because, as we said before, the dimension of a cell is given by the number of events that are currently executing concurrently.
– Axiom (A5') $\{\}^i\top \to \{\}i$ says something of the opposite (it looks at the upper levels instead of at the lower levels like the axiom before). The axiom says that if from the current cell it can move i levels up (i.e., can reach a cell of dimension i greater than the current cell) then is must be that from the current cell there can be started at least i different new events.

– Axiom (A9) $\langle\rangle^i\top \rightarrow \llbracket\rrbracket\langle\rangle\langle\rangle^i\top$ has two purposes. In the basic variant (for $i = 0$ it becomes $\llbracket\rrbracket\langle\rangle\top$) it says that in any cell, however one starts an event then one can also terminate an event. In the general form the axiom says that from some level i when going one level up (by starting an event) and then one level down (by terminating an event) we always end up on the same level i; i.e., we end in a cell of the same dimension like the cell that it started in.
– Axioms (A8) and (A8') intuitively say that if some cell has dimension greater than i then all the cells on that level have dimension greater than i.

For soundness of the axiomatic system it is enough to prove that the axioms (A5) to (A9') are valid. We start with axiom (A6) and assume $\mathcal{H}, q_n \models \langle\rangle[\,]\varphi$ for some $q_n \in Q_n$ and arbitrary n. This means that exists some $q_{n-1} \in Q_{n-1}$ s.t. $t_k(q_n) = q_{n-1}$ for some $1 \leq k \leq n$ with $\mathcal{H}, q_{n-1} \models [\,]\varphi$, and from this it means that for any $1 \leq l \leq n-1$, $\mathcal{H}, t_l(q_{n-1}) \models \varphi$. We need to show that $\mathcal{H}, q_n \models [\,]\langle\rangle\varphi$. This means that for any $m \neq k$ we have to find a $1 \leq m' \leq n-1$ s.t. $\mathcal{H}, t_{m'}(t_m(q_n)) \models \varphi$.[3] This is easy by applying the cubical law, considering wlog. $m < k$, $t_m(t_k(q_n)) = t_{k-1}(t_m(q_n))$. Thus, the $m' = k-1$ for which trivially $1 \leq k-1 \leq n-1$. From the assumption we showed that we have $\mathcal{H}, t_m(t_k(q_n)) \models \varphi$ and hence $\mathcal{H}, t_{k-1}(t_m(q_n)) \models \varphi$.

For axiom (A6') assume $\mathcal{H}, q_n \models \{\}\llbracket\rrbracket\varphi$ with $q_n \in Q_n$. This means that exists $q_{n+1} \in Q_{n+1}$ and $1 \leq k \leq n+1$ s.t. $s_k(q_{n+1}) = q_n$ and $\mathcal{H}, q_{n+1} \models \llbracket\rrbracket\varphi$. Which means that for any q_{n+2} with $s_l(q_{n+2}) = q_{n+1}$ for some $1 \leq l \leq n+2$, we have $\mathcal{H}, q_{n+2} \models \varphi$. We want to show that $\mathcal{H}, q \models \llbracket\rrbracket\{\}\varphi$ which reduces to showing that for an arbitrary $q'_{n+1} \in Q_{n+1}$ with $s_m(q'_{n+1}) = q$ for some $1 \leq m \leq n+1$, we can find an $1 \leq m' \leq n+2$ s.t. $s_{m'}(q_{n+2}) = q'_{n+1}$. This is done by the cubical laws: if $m \leq k$ consider $s_m(s_{k+1}(q_{n+2})) = s_k(s_m(q_{n+2}))$ and make $m' = k+1$ (it is clear that $m' \leq n+2$) and thus $\mathcal{H}, s_m(s_{m'}(q_{n+2})) \models \llbracket\rrbracket\{\}\varphi$ because we have said that $\mathcal{H}, s_k(s_m(q_{n+2})) \models \llbracket\rrbracket\{\}\varphi$; otherwise, if $k < m$ then consider the law $s_k(s_{m+1}(q_{n+2})) = s_m(s_k(q_{n+2}))$ (as $m+1 \leq n+2$) and take $m' = k$, thus having $\mathcal{H}, s_m(s_{m'}(q_{n+2})) \models \llbracket\rrbracket\{\}\varphi$ because $\mathcal{H}, s_k(s_{m+1}(q_{n+2})) \models \llbracket\rrbracket\{\}\varphi$.

For axiom (A7) assume $\mathcal{H}, q_n \models \{\}[\,]\varphi$ with $q_n \in Q_n$. This means that exists $q_{n+1} \in Q_{n+1}$ and $1 \leq k \leq n+1$ s.t. $s_k(q_{n+1}) = q$ and $\mathcal{H}, q_{n+1} \models [\,]\varphi$. Further, this implies that for any $1 \leq i \leq n+1$, $\mathcal{H}, t_i(q_{n+1}) \models \varphi$. We want to prove that $\mathcal{H}, q \models [\,]\{\}\varphi$ which amounts to showing that for some arbitrary $1 \leq m \leq n$ with $t_m(q) = q_{n-1}$ we can find an $1 \leq l \leq n$ and $q'_n \in Q_n$ s.t. $s_l(q'_n) = q_{n-1}$ and $\mathcal{H}, q'_n \models \varphi$. This is done with the cubical laws: if $m < k$ then consider the cubical law $t_m(s_k(q_{n+1})) = s_{k-1}(t_m(q_{n+1}))$ and set $l = k-1$ and $q'_n = t_m(q_{n+1})$ for which we know from above that $\mathcal{H}, t_m(q_{n+1}) \models \varphi$; otherwise if $k \leq m$ (which also means that $k \leq n$) then consider the cubical law $s_k(t_{m+1}(q_{n+1})) = t_m(s_k(q_{n+1}))$ and set $l = k$ and $q'_n = t_{m+1}(q_{n+1})$ (where $m+1 \leq n+1$) for which we know that $\mathcal{H}, t_{m+1}(q_{n+1}) \models \varphi$.

For axiom (A7') assume $\mathcal{H}, q_n \models \langle\rangle\llbracket\rrbracket\varphi$ with $q_n \in Q_n$. This means that exists $1 \leq k \leq n$ and q_{n-1} s.t. $t_k(q_n) = q_{n-1}$ and $\mathcal{H}, q_{n-1} \models \llbracket\rrbracket\varphi$, which means that for any q'_n with $s_i(q'_n) = q_{n-1}$ for some $1 \leq i \leq n$ we have $\mathcal{H}, q'_n \models \varphi$. We want to prove that $\mathcal{H}, q_n \models \llbracket\rrbracket\langle\rangle\varphi$ which amounts to showing that for some arbitrary q_{n+1}, with $s_m(q_{n+1}) = q_n$ for some $1 \leq m \leq n+1$, we can find an $1 \leq l \leq n+1$ and a q''_n s.t. $t_l(q_{n+1}) = q''_n$ and $\mathcal{H}, q''_n \models \varphi$. We use the cubical laws: if $k < m$ then consider the cubical law $t_k(s_m(q_{n+1})) = s_{m-1}(t_k(q_{n+1}))$ and set $l = k$ and $q''_n = t_k(q_{n+1})$ for which we have said before that $\mathcal{H}, t_l(q_{n+1}) \models \varphi$ because there is the s_{m-1} that

[3] We do not consider the k because the case for $m = k$ is trivial from the assumption above, where we know that for t_k and any t_l it is the case that $\mathcal{H}, t_l(t_k(q_n)) \models \varphi$.

reaches a cell which satisfies $\llbracket\rrbracket\varphi$; otherwise if $m \leq k$ then consider the cubical law $s_m(t_{k+1}(q_{n+1})) = t_k(s_m(q_{n+1}))$ and set $l = k+1$ and $q''_n = t_{k+1}(q_{n+1})$ for which it holds that $\mathcal{H}, t_l(q_{n+1}) \models \varphi$ because $\mathcal{H}, s_m(t_l(q_{n+1})) \models \llbracket\rrbracket\varphi$.

For axiom (A8) assume $\mathcal{H}, q_n \models \{\}\langle\rangle^i\top$. This means that exists q_{n+1} and $1 \leq j \leq n+1$ s.t. $s_j(q_{n+1}) = q_n$ and $\mathcal{H}, q_{n+1} \models \langle\rangle^i\top$. This means that the dimension of q_{n+1} is greater than i, i.e., $n+1 \geq i$. We want to prove that $\mathcal{H}, q_n \models \llbracket\rrbracket\langle\rangle^i\top$ which amount to showing that for any $q'_{n+1} \in Q_{n+1}$ with $s_j(q'_{n+1}) = q_n$ for some $1 \leq j \leq n+1$ we have $\mathcal{H}, q'_{n+1} \models \langle\rangle^i\top$. But we know from before that the dimension of q'_{n+1} is at least i; this means that we can go down at least i levels and on the lowest level any cell models $\top$. Hence we have $\mathcal{H}, q'_{n+1} \models \langle\rangle^i\top$.

For axiom (A8') we use a similar argument based on the semantics of $\langle\rangle$ and $[\,]$ this time.

For axiom (A9) assume $\mathcal{H}, q_n \models \langle\rangle^i\top$ which means that $n \geq i$. Even more, $\langle\rangle^i\top$ holds in any cell $q_n \in Q_n$ of dimension n. We need to prove that $\mathcal{H}, q_n \models \llbracket\rrbracket\langle\rangle\langle\rangle^i\top$. The proof is trivial when there is no q_{n+1} with $s_j(q_{n+1}) = q_n$. Therefore, we need to prove that for any q_{n+1} with $s_j(q_{n+1}) = q_n$, for some $1 \leq j \leq n+1$, $\mathcal{H}, q_{n+1} \models \langle\rangle\langle\rangle^i\top$. Because $q_{n+1} \in Q_{n+1}$ then it must have at least one t map that links it with some cell $q'_n \in Q_n$ on the lower level. In q'_n the formula $\langle\rangle^i\top$ holds and thus we finished the proof.

For axiom (A9') assume $\mathcal{H}, q_n \models \{\}\langle\rangle\langle\rangle^i\top$ which means that exists $q_{n+1} \in Q_{n+1}$ with $s_j(q_{n+1}) = q_n$ for some $1 \leq j \leq n+1$ s.t. $\mathcal{H}, q_{n+1} \models \langle\rangle\langle\rangle^i\top$. This means that $n+1 \geq i+1$ and thus $n \geq i$. Therefore, for any $q'_n \in Q_n$ the formula $\langle\rangle^i\top$ holds because we can go at least i levels down and find any cell satisfying $\top$, hence $\langle\rangle^i\top$ holds also in $q_n \in Q_n$.

Axioms (A5) and (A5') require inductive arguments and we left them at the end. For (A5) use induction on i, where the base case $\langle\rangle 1 \rightarrow \langle\rangle^1\top$ is trivial. Consider that $\mathcal{H}, q \models \langle\rangle i$ which means that there exist i different cells q^j with $1 \leq j \leq i$ which are the result of the application of a t map to q. Because t is a map it means that there exist at least i different maps t_j with $1 \leq j \leq i$. Therefore, q is of dimension at least i and any $t_j(q)$ is of dimension at least $i-1$. By the induction hypothesis we have that $\mathcal{H}, t_j(q) \models \langle\rangle^{i-1}\top$ and therefore $\mathcal{H}, q \models \langle\rangle\langle\rangle^{i-1}\top$, i.e., $\mathcal{H}, q \models \langle\rangle^i\top$.

For axiom (A5') we also use induction where the base case for $\{\}^1\top \rightarrow \{\}1$ is trivial. Consider that $\mathcal{H}, q \models \{\}^i\top$ which means that it exists $s_j(q') = q$ s.t. $\mathcal{H}, q' \models \{\}^{i-1}\top$. By the induction hypothesis we have that $\mathcal{H}, q' \models \{\}i-1$ which means that there are at least $i-1$ different q'' with $s_k(q'') = q'$ for some k. Pick one such q'' and note that q, q', q'' enter into a cubical law: if $k > j$ then $s_j(s_k(q'')) = q = s_{k-1}(s_j(q''))$, or if $k \leq j$ then $s_k(s_{j+1}(q'')) = q = s_j(s_k(q''))$. Any of the cases brings a cell different than q' (either $s_j(q'')$ or $s_{j+1}(q'')$) which is linked through an s map to q. Similarly, all the $i-1$ different q'' bring a different cell that links to q through an s map. Thus we have shown that q is linked with i different cells through one of their s maps.

For $i > 2$ we need to make sure that we really have new cells for each q''. Assume (by *reductio ad absurdum*) that for two different q''_k and q''_l, with $s_k(q''_k) = q' = s_l(q''_l)$ and wlog. $k > j$ and $l > j$, we have $s_j(q''_k) = s_j(q''_l) = q'_j$.[4] On another hand, because of the above cubical laws we have $s_{l-1}(q'_j) = s_{k-1}(q'_j)$ which can be possible only when $k = l$. This, together with the fact that both q''_k and q''_l are linked with some q''' by an s map, we can find them in the following cubical law: $s_k(s_{k+1}(q''')) = s_k(s_k(q'''))$. If this is the case then there are some other cubical laws applicable: $s_j(s_{k+1}(q''')) = q'_j = s_j(s_k(q'''))$. From this we get $s_k(s_j(q''')) = s_j(s_{k+1}(q''')) = s_j(s_k(q''')) =$

[4] Analogous arguments can be given for the other three cases depending on j, k, l.

$s_{k-1}(s_j(q'''))$. This means that from the cell $s_j(q''')$ there are two different s maps (i.e., s_{k-1} and s_k) that end in the same cell q'_j. This is not possible and hence we have the contradiction.

Proof (of Lemma 2.15). By induction on the structure of ψ.

Base case: $\psi = \phi \in \Phi_B$. From the definition we have $\mathcal{H}, q_n \models \phi$ iff $\phi \in \mathcal{V}(q_n)$ iff $\phi \in \lambda(q_n)$.

Inductive step: The case for the Boolean connectives follows easily from the properties on atoms. Finally we treat cases for the two modalities. Consider the during modality. The left to right direction is based on the canonicity of $\mathcal{H}$.

We prove $\mathcal{H}, q_n \models \{\}\varphi \Rightarrow \{\}\varphi \in \lambda(q_n)$. From the definition we know that $\exists q' \in Q_{n+1}$ and $\exists 0 \leq i \leq n+1$ s.t. $s_i(q') = q_n$ and $\mathcal{H}, q' \models \varphi$. From the induction hypothesis we have that $\mathcal{H}, q' \models \varphi$ iff $\varphi \in \lambda(q')$. Together with the canonicity of $\mathcal{H}$ we have that $\{\}\varphi \in \lambda(q_n)$. Proof finished.

For the right to left direction we use the saturation of $\mathcal{H}$. We prove $\{\}\varphi \in \lambda(q_n) \Rightarrow \mathcal{H}, q_n \models \{\}\varphi$. Using the saturation of $\mathcal{H}$ we have that $\exists q_{n+1} \in Q_{n+1}$ and $\exists 0 \leq i \leq n+1$ s.t. $s_i(q_{n+1}) = q_n$ and $\varphi \in \lambda(q_{n+1})$. By the induction hypothesis it implies that $\mathcal{H}, q_{n+1} \models \varphi$. Thus, by the definition we have that $\mathcal{H}, q_n \models \{\}\varphi$.

The proof for the $\langle\rangle$ modality is symmetric using the second conditions of canonicity and saturation of $\mathcal{H}$.

Proof (of Lemma 2.17).

```
function enrich(n,q,$\varphi$){
 $Q_{n+1}$ := $Q_{n+1} \cup \{q_{n+1}\}$; //fresh cell
 update map $s_{n+1}$ s.t. $s_{n+1}(q_{n+1}) = q$;
 label $q_{n+1}$ with $\lambda(q_{n+1}) = \{\varphi\} \cup \{\psi \mid [\![\,]\!]\psi \in \lambda(q)\}$;
 addSourceMaps(n+1,$q_{n+1}$,0,$\emptyset$);
 addTargetMaps(n+1,$q_{n+1}$,0,$\emptyset$);
}
function addSourceMaps($k$,$q$,$m$,$q'$){
 if(k>=1){
  $Q_{k-1}$ := $Q_{k-1} \cup \{q^1_{k-1}, \ldots, q^{k-1-m}_{k-1}\}$; //fresh cells
  for($l$=1 to m){
   update map $t_{k-l}$ s.t. $t_{k-l}(q) = t_{k-m}(t_{k-l+1}(q'))$;
  }
  for(i=k−1−m to 1){
   update map $s_i$ s.t. $s_i(q) = q^i_{k-1}$;
   update map $s_{k-1}$ s.t. $s_{k-1}(q^i_{k-1}) = s_i(s_k(q))$;
   label $q^i_{k-1}$ with $\lambda(q^i_{k-1}) = \lambda(s_k(q))$;
   addSourceMaps(k−1,$q^i_{k-1}$,k−1−m−i,q);
   addTargetMaps(k−1,$q^i_{k-1}$,0,$\emptyset$);
  }}}
function addTargetMaps($k$,$q$,$m$,$q'$){
 if(k >= 1){
  $Q_{k-1}$ := $Q_{k-1} \cup \{q^1_{k-1}, \ldots, q^{k-m}_{k-1}\}$; //fresh cells
  for($l$=0 to m−1){
   update map $t_{k-l}$ s.t. $t_{k-l}(q) = t_{k+1-m}(t_{k-l+1}(q'))$;
  }
  for(i=k−m to 1){
   update map $t_i$ s.t. $t_i(q) = q^i_{k-1}$;
   if(m=0 && i=k) label $q^i_{k-1}$ by $\lambda(q^i_{k-1}) = \{\psi \mid [\,]\psi \in \lambda(q)\}$;
   else label $q^i_{k-1}$ by $\lambda(q^i_{k-1}) = \lambda(t_k(q))$;
   if(k > 1){
    addTargetMaps(k-1, $q^i_{k-1}$, k-m-i, q);
    for(j=1 to k-1){ //add k-1 maps $s_j$ to $q^i_{k-1}$ cf. cubical laws
     if(j<i) update map $s_j$ s.t. $s_j(t_i(q)) = t_{i-1}(s_j(q))$;
     else    update map $s_j$ s.t. $t_i(s_{j+1}(q)) = s_j(t_i(q))$;
    }}}}}
```

For the sake of the presentation, in the algorithm we omitted the fact that after we finish adding all necessary formulas to a label (and before using this label further) we grow the label set to an atom cf. Lemma 2.13(iv). We can do this because we prove in the following that all the labels that we build are consistent sets. Similarly for the lifting construction in Lemma 2.18.

The proof has four stages: 1) we show that the enriched model is an extension of the old model (i.e., $\mathcal{H}' \rhd \mathcal{H}$); 2) we show that any set of formulas that the construction attaches as a label to a new cell is a consistent set, so that it can be grown to an atom, as required; 3) we show that the extended model $\mathcal{H}'$ is still canonical; 4) we show that $\mathcal{H}'$ is a model indeed, i.e., that all the maps are in place and all necessary cubical laws are respected. The proof makes use of the axioms (A7),(A7'), and (A9).

First of all remark that we do not change the initial shape or labels of the original $\mathcal{H}$; we only add fresh cells and fresh maps for these cells. This means that the enriched HDA extends $\mathcal{H}$.

Before proving that the enriched model $\mathcal{H}'$ is canonical, we should make sure that we indeed construct only consistent sets as labels. In this way Lemma 2.13(iv) can be applied to grow each set that we construct into an atom. There are only two places where we actually construct sets of formulas as labels; in all other places we just reuse labels of other existing cells which are already atoms.

The first set that we construct is in the enrich function itself. Assume that this set is not consistent, which means two cases: 1) $\psi_1 \wedge \cdots \wedge \psi_k \rightarrow \bot$, for $\psi_i \in \lambda(q_{n+1})$ and $\{\!\![\,]\!\!\}\psi_i \in \lambda(q)$ with $1 \leq i \leq k$, and 2) $\psi_1 \wedge \cdots \wedge \psi_k \rightarrow \neg\varphi$, for $\psi_i \in \lambda(q_{n+1})$, $\{\!\![\,]\!\!\}\psi_i \in \lambda(q)$, and $\{\}\varphi \in \lambda(q)$. For case 1) we know from modal logic that $\{\!\![\,]\!\!\}\psi_1 \wedge \cdots \wedge \{\!\![\,]\!\!\}\psi_k \rightarrow \{\!\![\,]\!\!\}(\psi_1 \wedge \cdots \wedge \psi_k)$ which, together with the assumption, it means that $\{\!\![\,]\!\!\}\psi_1 \wedge \cdots \wedge \{\!\![\,]\!\!\}\psi_k \rightarrow \{\!\![\,]\!\!\}\bot$. This means that $\{\!\![\,]\!\!\}\bot \in \lambda(q)$ which is a contradiction with the fact that $\lambda(q)$ is an atom containing an existential modality, namely, $\{\}\varphi$. For case 2) we follow a similar argument to obtain $\{\!\![\,]\!\!\}\psi_1 \wedge \cdots \wedge \{\!\![\,]\!\!\}\psi_k \rightarrow \{\!\![\,]\!\!\}(\psi_1 \wedge \cdots \wedge \psi_k) \rightarrow \{\!\![\,]\!\!\}\neg\varphi \rightarrow \neg\{\}\varphi$. But this is a contradiction because the atom $\lambda(q)$ already contains $\{\}\varphi$ and by Lemma 2.13(i) it cannot contain $\neg\{\}\varphi$.

The second set of formulas that the enrich construction adds is in the addTargetMaps. Assume that the set is inconsistent, i.e., $\psi_1 \wedge \cdots \wedge \psi_k \rightarrow \bot$ for $\psi_j \in \lambda(q^i_n)$ and $[\,]\psi_j \in \lambda(q_{n+1})$ with $1 \leq j \leq k$, where $t_i(q_{n+1}) = q^i_n$. By arguments similar as before we get $[\,]\psi_1 \wedge \cdots \wedge [\,]\psi_k \rightarrow [\,](\psi_1 \wedge \cdots \wedge \psi_k) \rightarrow [\,]\bot$ which would mean that $[\,]\bot \in \lambda(q_{n+1})$. But this is not possible because of the axiom (A9) which essentially says that there must be $\langle\rangle\top \in \lambda(q_{n+1})$, hence the contradiction. To be more precise, consider our initial cell $q_n \in Q_n$ of dimension n and hence $\mathcal{H}, q_n \models \langle\rangle^n\top$ which by axiom (A9) it means that $\{\!\![\,]\!\!\}\langle\rangle\langle\rangle^n\top \in \lambda(q_n)$. By the construction of the label of q_{n+1} we have $\langle\rangle\langle\rangle^n\top \in \lambda(q_{n+1})$. Because of this, we have an existential formula $\langle\rangle\top$ in the label that contains all the $[\,]\psi_j$ at any level, and hence at any depth of recursion calls to addTargetMaps. This means that at any recursive call the set of formulas that addTargetMaps builds is consistent.

We now prove that the enriched model $\mathcal{H}'$ is canonical. It is easier to treat the second canonicity condition and we do that now. Assume that for the enriched HDA the *second canonicity condition is broken*; i.e., assume $t_i(q) = q'$ for which $\varphi \in \lambda(q')$ and $\langle\rangle\varphi \notin \lambda(q)$, which is the same as $\neg\langle\rangle\varphi \in \lambda(q)$ cf. Lemma 2.13(i) because $\lambda(q)$ is an atom. We take cases after q. First, clearly, if $q, q' \in \mathcal{H}$ then the canonicity is assured by the statement of the lemma (i.e., $\mathcal{H}$ is canonical).

The rest is easy if we note that for new cells the only place where t_i maps are added is in the addTargetMaps function which is called in three places: for q_{n+1}, for cells added as targets for source maps s_i, and for cells added as targets of termination maps t_i. For all these cases the fact that $\neg\langle\rangle\varphi \in \lambda(q)$ implies that $[\,]\neg\varphi \in \lambda(q)$, and the construction of the label of $q' = t_i(q)$ implies that $\neg\varphi \in \lambda(q')$ which is impossible because $\lambda(q')$ is constructed as an atom and by our assumption $\varphi \in \lambda(q')$.

Assume now that for the enriched HDA the *first canonicity condition is broken*; i.e., assume it exists $s_i(q) = q'$ for which $\varphi \in \lambda(q)$ and $\{\}\varphi \notin \lambda(q')$. We now take cases after q. First, if $q, q' \in \mathcal{H}$ (i.e., is part of the old model that we have just enriched) then the assumption is violated because these cells respect the canonicity conditions (by the statement of the lemma).

Second, assume that $q = q_{n+1}$ and $q' = q_n^i$ is one of the cells added by the function addSourceMaps(n+1,q_{n+1}). Each of these cells is labeled by $\lambda(q_n^i) = \lambda(s_{n+1}(q_{n+1}))$. By our assumption, this means that $\{\}\varphi \notin \lambda(q_n)$, which, because $\lambda(q_n^i)$ is an atom it means that $\neg\{\}\varphi \in \lambda(q_n^i)$. By axiom (A2') it means that $\{\![\,]\!\}\neg\varphi \in \lambda(s_{n+1}(q_{n+1}))$. This means, by the construction of the label of q_{n+1}, that $\neg\varphi \in \lambda(q_{n+1})$. This is a contradiction as the atom $\lambda(q_{n+1})$ already has φ.

Third, assume q is one of the cells added by the addSourceMaps(n+1,q_{n+1}) at some recursion depth; i.e., $q = q_k^i$ for some $0 < k \leq n$ (i.e., $q_k^i \notin Q_0 \wedge q_k^i \notin Q_{n+1}$) and for some $1 \leq i \leq k$. Note that $s_k(q_k^i) = s_i(s_{k+1}(q_{k+1}^j))$ where $s_i(q_{k+1}^j) = q_k^i$ and that $s_i(s_{k+1}(q_{k+1}^j)) \in \mathcal{H}$ and $s_{k+1}(q_{k+1}^j) \in \mathcal{H}$. By assumption $\varphi \in \lambda(q_k^i) = \lambda(s_{k+1}(q_{k+1}^j))$ and $\{\![\,]\!\}\neg\varphi \in \lambda(q_{k-1})$ where $s_j(q_k^i) = q_{k-1}$ for some $1 \leq j \leq k$. But $\lambda(q_{k-1}) = \lambda(s_i(s_{k+1}(q_{k+1}^j)))$. Because it is part of the old $\mathcal{H}$, $s_i(s_{k+1}(q_{k+1}^j))$ respects the canonicity constraints, meaning that $\neg\varphi \in \lambda(s_{k+1}(q_{k+1}^j))$ which is impossible as the atom $\lambda(s_{k+1}(q_{k+1}^j))$ already contains φ by assumption.

Lastly, assume q is one of the cells added by addTargetMaps(); i.e., $q = q_k^i = t_i(q_{k+1})$ for some $1 \leq i \leq k$ and some recursion depth k. The initial assumption says that $s_j(q_k^i) = q'$, for some $1 \leq j \leq k$. Wlog. we assume $j < i$ and thus we are in the case when $q' = s_j(t_i(q_{k+1})) = t_{i-1}(s_j(q_{k+1}))$. This means that there is a source from q_{k+1} to a cell q'' which has a target to q'. But the cases above have treated q'' and q' and their target link respects the canonicity. Because of this and of the assumption $\{\![\,]\!\}\neg\varphi \in \lambda(q')$, implies $\langle\rangle\{\![\,]\!\}\neg\varphi \in \lambda(q'')$, call this *(*)*. From *(*)*, by axiom (A7') it means that $\{\![\,]\!\}\langle\rangle\neg\varphi \in \lambda(q'')$ *(**)*. Note that addTargetMaps() builds the same labels $\lambda(q_k^i)$ for all $i \leq k$ and by the construction of these labels it means that $[\,]\varphi \in \lambda(q_{k+1})$. From this and addSourceMaps() we have two cases: *(i)* $\{\![\,]\!\}[\,]\varphi \in \lambda(q'')$, *(ii)* $\{\}[\,]\varphi \in \lambda(q'')$. From *(i)* and *(**)* a simple reasoning gives that $\{\![\,]\!\}(\langle\rangle\neg\varphi \wedge [\,]\varphi) \rightarrow \{\![\,]\!\}\langle\rangle(\neg\varphi \wedge \varphi) \rightarrow \{\![\,]\!\}\bot$ by (A2'). Which means that $\{\![\,]\!\}\bot \in \lambda(q'')$, and thus $\bot \in \lambda(q_{n+1})$ which is a contradiction. From *(ii)* we have by (A7) that $[\,]\{\}\varphi \in \lambda(q'')$ and together with *(*)* we have $\langle\rangle(\{\}\varphi \wedge \{\![\,]\!\}\neg\varphi) \rightarrow \langle\rangle\{\}(\varphi \wedge \neg\varphi) \rightarrow \bot$ by axioms (A2) and (A2'). This is a contradiction as $\bot \notin \lambda(q'')$.

We should also make sure that we indeed construct a higher dimensional structure. A careful reading of the enriching construction should answer this question in affirmative.

We need to make sure that to each new cell we add all the s and t maps according to its dimension and that we link these maps correctly according to the cubical laws.

The enriching construction proceeds as follows. It takes our initial cell q and its dimension n and the formula that gives the D1 defect. It adds a new cell q_{n+1} of dimension one greater than q and links this with q through the s_{n+1} map. It labels the new cell s.t. the defect of q is repaired. The labeling is not important for our current argument but it is used in the argument for canonicity. To have the new cell q_{n+1} correctly added we need to add n more s maps (i.e., the s_i maps with $1 \leq i \leq n$) and $n+1$ more t maps to it. The s maps are added by the addSourceMaps and the t maps are added by the addTargetMaps.

Consider now the addSourceMaps function which take as arguments the cell to which it must add the maps and the dimension of this cell, together with two other arguments used for bookkeeping of the cubical laws that need to be added for each cell. More precisely, the m argument records how many cubical laws the q cell enters into. Note that this function (the same as addTargetMaps) adds maps only if the dimension of the cell is greater than 0, because, by definition, states in a HDA have no maps. addSourceMaps adds only $k-1$ maps to its cell argument because one s map has already been added before (e.g., for q_{n+1} we have added the map s_{n+1} and it remains to add the other maps from s_1 to s_n). All these maps link to new cells of dimension one lower (i.e., dimension $k-1$). Actually there are less new cells because some of the s maps must link to already existing cell so to respect the cubical laws. The m argument tells how many s maps should come only from cubical laws and hence, we add only $k-1-m$ new cells. The next loop adds these maps respecting the cubical laws; e.g., for the cell $q_n^{n-1} = s_{n-1}(q)$ we add the map $s_{n-1}(q_n^{n-1})$ as the result of $s_{n-1}(s_n(q))$ (which are cells that have already been added) because of the cubical law $s_{n-1}(s_n(q)) = s_{n-1}(s_{n-1}(q))$. In fact, for the cell q_n^1 each of its s maps links to some existing cell, thus no new cells are added.

Each of the $k-1-m$ new cells are linked with q by the corresponding s_i map. It is also added the s_{k-1} map (i.e., the map with greatest index among the $k-1$ maps that the cell needs). This is done so to respect the cubical laws $s_i(s_k(q)) = s_{k-1}(s_i(q))$. We now need to recursively add the required s and t maps for the new cell. We call the addSourceMaps for this cell q_{k-1}^i of dimension $k-1$ and, depending on the index i in the loop, we specify that $k-1-m-i$ maps should be added directly through the cubical laws and not by using new cells. We must also carry along the node q to which the cubical laws link. We also add the t maps for q_{k-1}^i by calling the addTargetMaps function.

The function addTargetMaps adds all the t maps of the cell (not one less as the addSourceMaps is doing). addTargetMaps also tries to respect the cubical laws first, and thus the m argument tells which maps come only from a cubical law like $t_i(t_j(q)) = t_{j-1}(t_i(q))$. For a cell q of dimension k addTargetMaps adds $k-m$ new cells of dimension $k-1$ and links each of these cells through a corresponding t_i map. For each new cell a recursive call to addTargetMaps is needed to add all the necessary t maps. The s maps of the new cells are added in the end taking care that all the cubical laws of the form $s_i(t_j(q)) = t_{j-1}(s_i(q))$ are respected. All these s maps are linked to cells which come from t maps that have been added by the addSourceMaps function before.

Proof (of Lemma 2.18).

The lifting construction is the following:

```
function lift(n,q,$\varphi$){
 addTargetMap(n,q,$\{\varphi\}$,$\lambda(q)$,$\emptyset$);
 addSourceMap(n,q,$\lambda(q)$,$\emptyset$);
 for(all cells $q' \neq q$){
  addTargetMap(n,$q'$,$\emptyset$,$\lambda(q')$,$\emptyset$);
  addSourceMap(n,$q'$,$\lambda(q')$,$\emptyset$);
 }}
function addTargetMap(k,q,S1,S2,S3){
 $Q_{k-1}$:=$Q_{k-1} \cup \{q_{k-1}\}$;  // fresh cell
 update map $t_{k+1}$ s.t. $t_{k+1}(q) = q_{k-1}$;
 label $q_{k-1}$ with $\lambda(q_{k-1})$ =S1∪{$\varphi$ | [ ]$\varphi$ ∈ S2∪ S3};
 for(i=1 to k){
  $r^i_{k-2}$:=addTargetMap(k−1,$s_i(q)$,$\emptyset$,$\lambda(s_i(q))$,$\emptyset$);
  $Q_{k-2}$:=$Q_{k-2} \cup \{r^i_{k-2}\}$;
  update map $s_i$ s.t. $s_i(t_{k+1}(q)) = r^i_{k-2}$;
  $\lambda(q_{k-1})$:=$\lambda(q_{k-1}) \cup \{\varphi \mid \{\!]\varphi \in \lambda(r^i_{k-2})\}$;  // update label
 }
 for(i=1 to k){
  $q^i_{k-2}$:=addTargetMap(k−1,$t_i(q)$,$\emptyset$,$\lambda(t_i(q))$,$\lambda(q_{k-1})$);
  $Q_{k-2}$:=$Q_{k-2} \cup \{q^i_{k-2}\}$;
  update map $t_i$ s.t. $t_i(t_{k+1}(q)) = q^i_{k-2}$;
 }
 return $q_{k-1}$;
}
function addSourceMap(k,q,S1,S2){
 $Q_{k-1}$:=$Q_{k-1} \cup \{q_{k-1}\}$;  // fresh cell
 update map $s_{k+1}$ s.t. $s_{k+1}(q) = q_{k-1}$;
 label $q_{k-1}$ with $\lambda(q_{k-1}) = \{\{\}\varphi \mid \varphi \in$ S1∪ S2};
 for(i=1 to k){  //add a fresh $s_k$ map to each $t_i(q)$
  $r^i_{k-2}$:=addSourceMap(k−1,$t_i(q)$,$\lambda(t_i(q))$,$\emptyset$);
  $Q_{k-2}$:=$Q_{k-2} \cup \{r^i_{k-2}\}$;
  update map $t_i$ s.t. $t_i(s_{k+1}(q)) = r^i_{k-2}$;
  $\lambda(q_{k-1})$:=$\lambda(q_{k-1}) \cup \{\langle\rangle\varphi \mid \varphi \in \lambda(r^i_{k-2})\}$;  //update label
 }
 for(i=1 to k){  //add a fresh $s_k$ map to each $s_i(q)$
  $q^i_{k-2}$:=addSourceMap(k−1,$s_i(q)$,$\lambda(s_i(q))$,$\lambda(q_{k-1})$);
  $Q_{k-2}$:=$Q_{k-2} \cup \{q^i_{k-2}\}$;
  update map $s_i$ s.t. $s_i(s_{k+1}(q)) = q^i_{k-2}$;
 }
 return $q_{k-1}$;
}
```

The proof has the same four stages as the proof of Lemma 2.17: 1) showing that $\mathcal{H}' \triangleright \mathcal{H}$; 2) showing that the labeling introduces only consistent sets of formulas, so that they can be grown to atoms; 3) showing that $\mathcal{H}'$ is still canonical; 4) showing that $\mathcal{H}'$ is a model indeed, i.e., that all the maps are in place and all necessary cubical laws are respected. The proof makes use of the axioms (A5), (A6), (A6'), (A7'), (A8'), and (A9').

Note that the algorithm finishes with a completely new layer of cells denoted Q_{-1}; in the end of the construction we have to rename all the layers Q_i into Q_{i+1} to make justice to the cells that reside there which have now dimension $i+1$ as we added one s and one t map to each. Also note that the construction terminates iff q is in a hypercube of finite dimension and in this case we ignore all the cells outside this cube. (The construction always terminates when we use it in the repair lemma 2.19.) Clearly the two functions do not change the labels nor the shape of the old $\mathcal{H}$ and hence the lifted $\mathcal{H}'$ extends $\mathcal{H}$. We continue to show that $\mathcal{H}'$ is canonical.

Generally the proof of canonicity will flow nicely from the labeling, but first we must make sure that the labels introduced by the construction are consistent sets (here is where the axioms are put to work).

Each application of any of the two functions adds a single new cell and builds its label as well as its s and t maps. Consider the addTargetMap function; there are two places where the label is changed, first is in the beginning where the three input sets are used, and second is in the first loop where in each iteration the label is updated. The set S1 is not empty only when the function is applied to the initial cell q from the statement of the lemma. The lemma assumes that $q \in Q_n$ is of dimension n, denote it q_n for this part of the proof, and it contains $\langle\rangle\varphi \in \lambda(q_b)$ for which none of its s maps contain φ. This means that if before we could write $\langle\rangle n \in \lambda(q_n)$ now we need to write $\langle\rangle n + 1$. Because of axiom (A5) and Lemma 2.13(ii) it means that $\langle\rangle^{n+1}\top \in \lambda(q_n)$. (As a side remark, we use Lemma 2.13(ii) tacitly in many places during the proofs of the two constructions lemmas.)

The first call to addTargetMap(n,q_n,$\{\varphi\}$,$\lambda(q_n)$,$\emptyset$) makes use of only sets S1 and S2 and constructs the set $\{\varphi\} \cup \{\psi \mid [\,]\psi \in \lambda(q_n)\}$. This set is associated to $q_{n-1} = t_{n+1}(q_n)$. The proof is easy for this case and uses arguments as in the proof before: if we assume $\psi_1 \wedge \cdots \wedge \psi_k \rightarrow \bot$ then we get that $[\,]\bot \in \lambda(q_n)$ which is a contradiction as $\lambda(q_n)$ is an atom containing $\langle\rangle\varphi$; if we assume $\psi_1 \wedge \cdots \wedge \psi_k \rightarrow \neg\varphi$ then we get that $[\,]\neg\varphi \in \lambda(q_n)$ which is again a contradiction.

The second call to addTargetMap is made for each s map of a cell q (in the first loop of the body of the addTargetMap) and it uses only the set S2. This means that it labels a cell $q_{n-1} = t_{n+1}(q)$ with a set $\{\psi \mid [\,]\psi \in \lambda(q)\}$. Assume $\psi_1 \wedge \cdots \wedge \psi_k \rightarrow \bot$ which means that $[\,]\bot \in \lambda(q)$. This is a contradiction because $\lambda(q)$ is an atom and it contains at least one diamond formula. This is because q has dimension at least 1 (as it has at least one t map) and we show that any cell of dimension n, with $n \geq 1$, has a formula $\langle\rangle^n\top \in \lambda(q)$. We showed before that the topmost cell q_n has the formula $\langle\rangle^{n+1}\top$ in its label and hence it is of dimension $n+1$. This means that any cell reached through one of its t maps will have the formula $\langle\rangle^n\top$ because of axiom (A8') which says that $\langle\rangle\langle\rangle^n\top \rightarrow [\,]\langle\rangle^n\top$ it means that $[\,]\langle\rangle^n\top \in \lambda(q_n)$ and by the construction of their labels it means that $\langle\rangle^n\top \in \lambda(t_j(q_n))$. This holds for any cell reached through any number of applications of t maps. On the other hand, the cells reached through an s map from q_n, by canonicity, they contain $\{\}\langle\rangle^{n+1}\top$, which, by axiom (A9') it means that $\langle\rangle^n\top \in \lambda(s_j(q_{n)})$.

It remains to see that with each iteration of the first loop the updated label remains a consistent set. This update is necessary when we are trying to respect the cubical laws of the form $s_i(t_{k+1}(q)) = t_k(s_i(q))$. The proof of this part follows an inductive argument, where the basis was just proven above and the inductive case is for some i iteration, where we consider that the label is a consistent set (and all the other labels that the construction uses have been built already and, hence, are atoms). Assume that for some $\{\![\,]\!\}\psi \in \lambda(t_k(s_i(q)))$ there has already been added the $\neg\psi$ to $\lambda(t_{k+1}(q))$. This has happened in two cases: first if $\neg\psi$ comes from $\lambda(q)$, i.e., $[\,]\neg\psi \in \lambda(q)$ which by canonicity it means that $\{\}[\,]\neg\psi \in \lambda(s_i(q))$. On the other hand we also have that $\langle\rangle\{\![\,]\!\}\psi \in \lambda(s_i(q)) \overset{(A7')}{\rightarrow} \{\![\,]\!\}\langle\rangle\psi \in \lambda(s_i(q))$. Together with the above it means that $\{\}([\,]\neg\psi \wedge \langle\rangle\psi) \rightarrow \{\}\langle\rangle(\neg\psi \wedge \psi) \overset{(A2),(A2')}{\longrightarrow} \bot \in \lambda(s_i(q))$ which is a contradiction with the fact that $\lambda(s_i(q))$ is an atom. The second case is when $\neg\psi$ has been added in a previous iteration, i.e., $\{\![\,]\!\}\neg\psi \in \lambda(s_j(t_{k+1}(q)))$ with $1 \leq j < i$. But this means that each of these two cells must have at least one s map and enter the cubical law

$s_j(s_i(t_{k+1}(q))) = s_{i-1}(s_j(t_{k+1}(q))) = q''$. By the canonicity of these lower cells we have that $\{\}\llbracket\rrbracket\neg\psi \in \lambda(q'')$ and $\{\}\llbracket\rrbracket\psi \in \lambda(q'')$. From axiom (A6') we have that $\llbracket\rrbracket\{\}\psi \in \lambda(q'')$ and thus $\llbracket\rrbracket\{\}\psi \wedge \{\}\llbracket\rrbracket\neg\psi \to \{\}(\{\}\psi \wedge \llbracket\rrbracket\neg\psi) \to \{\}\{\}(\psi \wedge \neg\psi) \overset{(A2)}{\to} \bot \in \lambda(q'')$ which is a contradiction.

The application of addTargetMap in the second loop uses the S3 set also and we are looking at cubical laws of type $t_i(t_{k+1}(q)) = t_n(t_i(q))$ where $S2 = \lambda(t_i(q))$ and $S3 = \lambda(t_{k+1}(q))$. Assume, for the sake of contradiction, that we have $[\,]\psi \in \lambda(t_{k+1}(q))$ and $[\,]\neg\psi \in \lambda(t_i(q))$. By canonicity it means that $\langle\rangle[\,]\psi \in \lambda(q)$ and $\langle\rangle[\,]\neg\psi \in \lambda(q)$ and from axiom (A6) we have $[\,]\langle\rangle\neg\psi \in \lambda(q)$. This means that $[\,]\langle\rangle\neg\psi \wedge \langle\rangle[\,]\psi \to \langle\rangle(\langle\rangle\neg\psi \wedge [\,]\psi) \to \langle\rangle\langle\rangle(\psi \wedge \neg\psi) \overset{(A2')}{\to} \bot \in \lambda(q)$ which is a contradiction.

For the labels added by the addSourceMap function it is easy to see that no inconsistencies can appear. In the beginning of the function, the label is constructed by applying the $\{\}$ modality to the formulas from S2 and S3. This cannot give inconsistencies. Then, in the first loop of the function, the label is updated by adding formulas that apply the $\langle\rangle$ modality, and again we cannot get inconsistencies.

Next we show that $\mathcal{H}'$ is canonical. Assume that for the lifted HDA the *second canonicity condition is broken*; i.e., consider $q \in Q_n$ and assume $t_i(q) = q'$ for which $\varphi \in \lambda(q')$ and $\langle\rangle\varphi \notin \lambda(q)$, which is the same as $\neg\langle\rangle\varphi \in \lambda(q)$. We take cases after q.

First, clearly, if $q, q' \in \mathcal{H}$ (meaning that $1 \leq i \leq n-1$) then the canonicity is assured by the statement of the lemma (i.e., $\mathcal{H}$ is canonical).

Second, $q \in \mathcal{H}$ and q' is added by addTargetMap as the new cell linked to q by $t_n(q) = q'$. Now we take sub-cases depending on where does the φ formula come from.

– If $\varphi \in S1$; this is the case when q is the initial cell from the statement of the lemma and hence it cannot be that $\neg\langle\rangle\varphi \in \lambda(q)$.
– If $\varphi \in \{\varphi \mid [\,]\varphi \in S2\}$ then $[\,]\varphi \in \lambda(q)$ and the assumption says that $[\,]\neg\varphi \in \lambda(q)$. This is a contradiction as $[\,]\varphi \wedge [\,]\neg\varphi \to [\,](\varphi \wedge \neg\varphi) \to [\,]\bot \in \lambda(q)$ which is not possible because, as we showed before, $\lambda(q)$ contains at least one existential formula, i.e., $\langle\rangle^k\top$, where k is the dimension of q.
– If $\varphi \in \{\varphi \mid [\,]\varphi \in S3\}$ then q' is added by the second call to addTargetMap, which means that we are respecting the cubical laws $t_i(t_{k+1}(q_{k+1})) = t_k(t_i(q_{k+1}))$, for $1 \leq i \leq k$ and for some q_{n+1} for which our $q = t_i(q_{k+1})$. Then by the construction of the label it means that $[\,]\varphi \in \lambda(t_{k+1}(q_{k+1}))$ which by the canonicity of these upper cells it means that $\langle\rangle[\,]\varphi \in \lambda(q_{k+1})$. By axiom (A6) it means that $[\,]\langle\rangle\varphi \in \lambda(q_{k+1})$ and thus, by the canonicity it means that $\langle\rangle\varphi \in \lambda(q)$ which is a contradiction with our initial assumption as the labels are atoms and hence $\neg\langle\rangle\varphi$ cannot be in the label $\lambda(q)$.
– Lastly, assume that φ is one of the formulas accumulated in the label of q' as a result of the first loop of addTargetMap. This means that we are respecting the cubical laws $s_i(t_{k+1}(q)) = t_k(s_i(q))$ and $\llbracket\rrbracket\varphi \in \lambda(s_i(q')) = \lambda(s_i(t_{k+1}(q))) = \lambda(t_k(s_i(q)))$. By canonicity of the other cells it means that $\langle\rangle\llbracket\rrbracket\varphi \in \lambda(s_i(q))$ which by axiom (A7') it means that $\llbracket\rrbracket\langle\rangle\varphi \in \lambda(s_i(q))$. By canonicity again it means that $\langle\rangle\varphi \in \lambda(q)$ which is again a contradiction with our initial assumption.

Third, both q and q' are newly added by addTargetMap, meaning that we are looking at the second loop. The proof is the same as before as the construction of the label and axiom (A6) do all the work.

Forth, both q and q' are newly added by addSourceMap, which means that we are in the first loop of addSourceMap and there exists a q_{k+1} with $s_{k+1}(q_{k+1}) = q$ and

$t_i(s_{k+1}(q_{k+1})) = q' = s_k(t_i(q_{k+1}))$ for some i. By the construction of the label of $s_{k+1}(q_{k+1})$, i.e., $\lambda(q)$, we have that for our formula $\varphi \in \lambda(q')$ there exists $\langle\rangle\varphi \in \lambda(q)$ because these are added in the label of q in the i step of the loop.

Assume that for the lifted HDA the *first canonicity condition is broken*; i.e., consider $q \in Q_n$ and assume $s_i(q) = q'$ for which $\varphi \in \lambda(q)$ and $\{\}\varphi \notin \lambda(q')$, which is the same as $\neg\{\}\varphi \in \lambda(q')$, or, by axiom (A4), $\{\!\!\{\}\!\!\}\neg\varphi \in \lambda(q')$. We again take cases after q.

Consider that $q \in \mathcal{H}$ and q' is added by the function addSourceMap. This may be done either in the first or in the second loop, but in any of the cases the construction of the labels ensures that if $\varphi \in \lambda(q)$ then $\{\}\varphi \in \lambda(q')$. The same holds for the case when both q and q' are newly added by the second call to addSourceMap (in the second loop).

Consider the case when both q and q' are newly added by the first call to the function addTargetMap. Our initial assumption says that $\{\!\!\{\}\!\!\}\neg\varphi \in \lambda(q')$ which means, by the iterative construction of the label of q in the loop, that $\neg\varphi \in \lambda(q)$ which is a contradiction with our initial assumption that $\varphi \in \lambda(q)$.

By now we are sure that the labeling of $\mathcal{H}'$ is canonical. Now we show that the lifting constructs indeed a HDA. This means that we must make sure that all the cells have the right number of s and t maps and that all the cubical laws are respected.

The lift function takes as input the reference cell q and its dimension n together with the formula φ that causes the defect (i.e., $\langle\rangle\varphi \in \lambda$). Then the function adds one t map and one s map to q by calling addTargetMap and addSourceMap respectively. These two functions add one new cell and link it with either a t or an s map. All other cells that are connected to q must also be lifted, which is done in the loop of the lift function.

Consider now the addTargetMap which takes as arguments the cell q (and its dimension k) to which the new t map needs to be added. It also takes three sets of formulas which are used to construct the label of the new cell. We do not discuss here the labeling because we did this before. The rest of the proof is concerned with the geometric structure of the extended $\mathcal{H}'$.

The addTargetMap function adds a new cell q_{k-1} of dimension one lower than that of the input cell q. It adds the new t_{k+1} map to q, which is the map with the largest index (i.e., the new index showing that the q cell has now dimension one greater, $k+1$). The first loop does two operations. First it lifts all the old cells linked to q by an s map by adding one t map to each. Then, all these cells enter under new cubical laws that involve the s maps of the newly added q_{k-1} cell. In this way we add all the necessary s maps of q_{k-1} and also lift all the cells linked by an s map to q and respect cubical laws $s_i(t_{k+1}(q)) = t_k(s_i(q))$. In the second loop we add the new t_k map to each old cell linked to q by a t_i map. At the same time we add all the t maps for the new q_{k-1} cell and link these through the cubical laws $t_i(t_{k+1}(q)) = t_k(t_i(q))$.

The construction goes recursively at lower levels until reaching cells of dimension 0. These are the last cells lifted to have dimension 1. Here the recursion stops.

Consider now the similar function addSourceMap which adds one s map to the input cell q of dimension k to make it now of dimension $k+1$. Therefore, it adds the map $s_{k+1}(q) = q_{k-1}$. In the first loop the function adds the new s_k maps to all the old cells linked to q by a t map. This finishes what we started in the second loop of addTargetMap, i.e., finishes lifting all the $t_i(q)$ cells. It also takes care to respect all the new cubical laws $t_i(s_{k+1}(q)) = s_k(t_i(q))$ and, hence, to add the t_i maps to q_{k-1}.

The second loop complements what we started in the first loop of addTargetMap. We finish adding the s_k maps to all the $s_i(q)$ cells. It also adds all the s maps to q_{k-1} and respects the new cubical laws $s_i(s_{k+1}(q)) = s_k(s_i(q))$.

Proof (of Proposition 3.2). We prove the double implication

$$\mathcal{H}, q \models \neg\langle\rangle\top \;\rightarrow\; \{\!\!\{\}\!\!\}\{\!\!\{\}\!\!\}\bot \quad \text{iff} \quad \mathcal{H} \text{ is as in Lemma 3.1,}$$

for some *HDA* $\mathcal{H}$ and any $q \in Q$ a cell of any dimension. For the *if* direction if $q \in Q_1$ then $\mathcal{H}, q \not\models \neg\langle\rangle\top$ and hence the axiom holds trivially. When $q \in Q_0$ then $\mathcal{H}, q \models \neg\langle\rangle\top$ and thus it must be that $\mathcal{H}, q \models \{\!\!\{\}\!\!\}\{\!\!\{\}\!\!\}\bot$. This is true because for any $q' \in Q_1$ with $s_1(q') = q$ it is the case that $\mathcal{H}, q' \models \{\!\!\{\}\!\!\}\bot$ because there are no $q'' \in Q_2$ cf. Lemma 3.1.

For the *only if* direction consider a $\mathcal{H}$ for which the axiom holds; we need to show that any Q_n with $n > 1$ is empty. Assume the opposite, that there exists $q_n \in Q_n$ with $n > 1$. This means that there is a sequence $s_1(\dots s_i(q_n)) = q_0$ of source maps that ends in a cell $q_0 \in Q_0$ of dimension 0. But $\mathcal{H}, q_0 \models \neg\langle\rangle\top \rightarrow \{\!\!\{\}\!\!\}\{\!\!\{\}\!\!\}\bot$, which means that there cannot be this sequence of source maps unless q_n is of dimension at most 1. This is a contradiction and hence the proof is finished.

Proof (of Theorem 3.3). First we check that we capture exactly the semantics of standard modal logic; $\mathcal{H}, q_0 \models \Diamond\varphi$ iff $\mathcal{H}, q_0 \models \{\}\langle\rangle\varphi$ iff $\exists q' \in Q_1$ s.t. $s_1(q') = q_0$ and $\mathcal{H}, q' \models \langle\rangle\varphi$ iff $\exists q_0' \in Q_0$ s.t. $t_1(q') = q_0'$ and $\mathcal{H}, q_0' \models \varphi$. This is the same as $\exists q_0' \in Q_0$ reached in “one transition” from q_0 and $\mathcal{H}, q_0' \models \varphi$. (We go only through one transition cell $q' \in Q_1$.)

Clearly, with the axiom of Proposition 3.2, $\mathcal{H}, q_n \not\models \Diamond\varphi$ for any $q_n \in Q_n$ for any $n \geq 1$. Therefore, $\Diamond\varphi$ makes sense only interpreted in states from Q_0.

Second we check that the axioms of standard modal logic for $\Diamond$ hold in our axiomatic system. Clearly $\Diamond\bot \leftrightarrow \bot$; just apply (A2') and then (A2) to $\{\}\langle\rangle\bot$. It is easy to see that $\Box\varphi \leftrightarrow \neg\Diamond\neg\varphi$ as $\neg\{\}\langle\rangle\neg\varphi \overset{(A4)}{\leftrightarrow} \{\!\!\{\}\!\!\}\neg\langle\rangle\neg\varphi \overset{(A4')}{\leftrightarrow} \{\!\!\{\}\!\!\}[\,]\varphi$ and the semantic of $\Box\varphi$ is the right one, i.e., for any $q_0' \in Q_0$, reached through some transition $q' \in Q_1$, is the case that $\mathcal{H}, q_0' \models \varphi$. We prove now that $\Diamond(\varphi \vee \varphi') \leftrightarrow \Diamond\varphi \vee \Diamond\varphi'$. This is because $\{\}\langle\rangle(\varphi \vee \varphi') \overset{(A3')}{\leftrightarrow} \{\}(\langle\rangle\varphi \vee \langle\rangle\varphi') \overset{(A3)}{\leftrightarrow} \{\}\langle\rangle\varphi \vee \{\}\langle\rangle\varphi' \overset{def}{\leftrightarrow} \Diamond\varphi \vee \Diamond\varphi'$.

It is easy to see how we recover the corresponding inference rule for $\Diamond$. We thus have all the axiomatic system of standard modal logic and the proof is finished.

Proof (of Proposition 3.6). Essential for the proof is the fact that $\overline{\mathcal{U}}$ is interpreted over restricted $HDAs$ which model Kripke structures. Precisely, they have only cells of dimension 0 (the states) and 1 (the transitions), and moreover, we know which are states because the formula $\neg\langle\rangle\top$ holds in all and only the cells of dimension 0. Therefore, the right formula of the $\overline{\mathcal{U}}$ is evaluated only in states because $(\varphi' \wedge \neg\langle\rangle\top)$ can never hold in a cell of dimension greater than 0. Moreover, the transitions are not important for valuating the φ because the formula $\langle\rangle\top$ is always true in a transition (because any transition has a target state). On the other hand the formula $\langle\rangle\top$ is never true in a state and hence the φ has to be true so that the whole left part of the until to hold.

For this proof we only concentrate on showing that the semantics of the $\overline{\mathcal{U}}$ corresponds to the well known LTL semantics. Thus, we should have $\mathcal{H}, q_0 \models \varphi\overline{\mathcal{U}}\varphi'$ iff exists a finite sequence $q_0^1, \dots, q_0^k$ with $q_0^1 = q_0$, $\mathcal{H}, q_0^k \models \varphi'$, $\mathcal{H}, q_0^i \models \varphi$ for all $1 \leq i < k$, and for any $1 < i \leq k$ q_0^i is reachable through a single transition from q_0^{i-1}. We have $\mathcal{H}, q_0 \models (\varphi \vee \langle\rangle\top)\,\mathcal{U}\,(\varphi' \wedge \neg\langle\rangle\top)$ and by the semantics of $\mathcal{U}$ from Definition 3.5 we know that $\exists\pi$ a path in the Kripke structure (i.e., going only through cells of dimension 0 or 1; which means is of the form $q_0, q_1, q_0', \dots$) s.t. $st(\pi) = q_0 \wedge en(\pi) = q'$, $\mathcal{H}, q' \models (\varphi' \wedge \phi_0)$, and $\forall q'' \in \pi, q'' \neq q'$ then $\mathcal{H}, q'' \models (\varphi \vee \langle\rangle\top)$. Clearly $q' \in Q_0$ because $\neg\langle\rangle\top$ must hold in q' and hence φ' holds in a state, i.e., $\mathcal{H}, q' \models \varphi'$. It remains to show that in all q'' which are states (i.e., those $q'' \in Q_0$) we have that $\mathcal{H}, q'' \models \varphi$. But we know that $\mathcal{H}, q'' \not\models \langle\rangle\top$ because q'', being a cell of dimension 0, has no t map. Therefore, using the before we have that $\mathcal{H}, q'' \models \varphi$.