The Undecidability of Arbitrary Arrow Update Logic

Hans van Ditmarsch[a], Wiebe van der Hoek[b], Louwe B. Kuijer[b,a,*]

[a] *LORIA, Campus scientifique, BP 239, 54506 Vandoeuvre-lès-Nancy Cedex, France*
[b] *Department of Computer Science, University of Liverpool, Ashton Building, Ashton Street, Liverpool L69 3BX, United Kingdom*

Abstract

Arbitrary Arrow Update Logic is a dynamic modal logic with a modality to quantify over arrow updates. Some properties of this logic have already been established, but until now it remained an open question whether the logic's satisfiability problem is decidable. Here, we show by a reduction of the tiling problem that the satisfiability problem of Arbitrary Arrow Update Logic is co-RE hard, and therefore undecidable.

Keywords: Modal Logic, Dynamic Epistemic Logic, Update Logics, Undecidability, Satisfiability

1. Introduction

Update Logics are logics that provide an object language in which one can reason about the effect of changes to a model for that language. Such an underlying model is usually a Kripke model, equipped with a set of states and some relations between them. One of the most prominent examples of updates relate to the incorporation of new *information*. This field of studies has become popular as *Dynamic Epistemic Logic (DEL)* [7] in the past decades. In epistemic logic, states in a Kripke model represent a description of the world, and the relations represent 'possibility' (for belief) or 'indistinguishability' (for knowledge) relations. We say that $\Box\varphi$ is true in state s in model $\mathcal{M}$, written $\mathcal{M}, s \models \Box\varphi$, if for all t, if $(s,t) \in R(a)$ then $\mathcal{M}, t \models \varphi$; that is, if in all states that are indistinguishable for agent a, formula φ holds.

Keeping this epistemic setting in mind for the moment, *Public Announcement Logic (PAL)* [12, 4], studies updates in which certain states of $\mathcal{M}$ are removed: $[\varphi]\psi$ means that after the announcement φ (which is interpreted as the operation in which only the φ-states are retained in the model), ψ holds. For example, if φ means "the door is locked" and ψ means "agent a believes

*Corresponding author
Email addresses: `hans.van-ditmarsch@loria.fr` (Hans van Ditmarsch), `wiebe.van-der-hoek@liverpool.ac.uk` (Wiebe van der Hoek), `Louwe.Kuijer@liverpool.ac.uk` (Louwe B. Kuijer)

she cannot access the room", then $[\varphi]\psi$ means "after it is announced that the door is locked, agent a will believe that she cannot access the room."

In *Arrow Update Logic (AUL)* [11], updates take the form of removing some *access* between states: $[(\varphi, a, \chi)]\psi$ denotes that if we only keep connections between two states if they are labelled a and go from a φ state to a χ state, ψ will hold. For example, for the same meaning of φ and ψ as above, $[(\varphi, a, \varphi)]\psi$ means "if whenever the door is locked (φ) agent a is told so ($\varphi = \chi$), then she (correctly) believes that she cannot access the room (ψ)".

Arrow updates are more powerful than public announcements; unlike public announcements, arrow updates can be used to model situations where different agents gain different information. For example, a might be told whether the door is locked while b is left in the dark on the matter. However, arrow updates can only *remove* arrows, they cannot *add* them. As a result, arrow updates can only be used to model situations where the amount of uncertainty decreases. If we want to model situations where the amount of uncertainty increases we will need to use an even more powerful kind of update. Among these more powerful kinds of updates, the most commonly used are *action models* [4]. Action models can, for example, be used to model the event where, from agent b's perspective, it is possible that a is told about whether the door is locked but it is also possible that a is not told.

The logics using public announcements, arrow updates and action models are called Public Announcement Logic (*PAL*), Arrow Update Logic (*AUL*) and Action Model Logic (*AML*)[1], respectively.

For each of these logics there is also an "arbitrary" version: for *PAL* there is Arbitrary Public Announcement Logic (*APAL*) [3], for *AUL* there is Arbitrary Arrow Update Logic (*AAUL*) [8] and for *AML* there is Arbitrary Action Model Logic (*AAML*) [10]. These "arbitrary" logics contain an operator that quantifies over their non-arbitrary counterpart. So in *APAL* we have $[!]\psi$ if and only if $[\varphi]\psi$ holds for every *PAL* formula φ, in *AAUL* we have $[\updownarrow]\psi$ if and only if $[U]\psi$ for every *AUL* update U and in *AAML* we have $[\times]\psi$ if and only if $[M]\psi$ for every *AML* action model M.

The logics *PAL*, *AUL*, and *AML* are equally expressive [4, 11]. The arbitrary versions of the logics are not equally expressive, however. Under reasonable assumptions about the number of agents, the logics *APAL* and *AAUL* are incomparable in expressivity [8], and they are both strictly more expressive than *AAML* [3, 8], since the latter logic is no more expressive than basic modal logic [10].

Two other logics that are similar to these "arbitrary" logics are Group Announcement Logic (*GAL*) [1] which allows quantification over a specific type of public announcements that are made by a group of agents, and Coalition Announcement Logic (*CAL*) [2] which allows us to ask whether there is some announcement for a group G such that ψ becomes true regardless of what all

[1] *AML* is also sometimes referred to as Dynamic Epistemic Logic (*DEL*), but here we reserve that name for the family of update logics of which *AML* is one.

agents outside of G announce.

It is important to realise that the relevance of this kind of updates goes beyond the realm of epistemic interpretations. In normative reasoning for instance, eliminating (bad) states enables one to reason about deontically 'better' situations, and eliminating (bad) transitions enforces 'better' behaviour. For more on the epistemic and normative interpretations of updates, see [8, Section 2].

In this paper, we focus on *AAUL*. So we consider the operator $[\updownarrow]$ that quantifies over all arrow updates.

Several technical results regarding *AAUL* were established in [8]. Specifically, the following results were proven. *Expressivity*: [8] shows that, under some mild assumptions, *APAL* and *AAUL* are incomparable over the class of all Kripke models. A case in which *AAUL* is more expressive than *APAL* is also identified. Successively, *AAUL* is compared to a number of other logics: it is established that *AAUL* is incomparable to epistemic logic with common knowledge, but more expressive than *PAL*. It is known that basic epistemic logic, public announcement logic *PAL*, arbitrary action model logic *AAML*, and refinement modal logic [6] are all equally expressive. As a corollary of this result we therefore also have that *AAUL* is more expressive than *AAML*. *Model Checking*: [8] shows that the model checking problem for *AAUL* is PSPACE-complete. *Axiomatisation*: An (infinitary) proof system for *AAUL* is introduced in [8] and its soundness and correctness (with respect to the set of intended models) is proven.

The question we address for *AAUL* in this paper regards its *decidability*. For some of the 'arbitrary' logics mentioned above, namely *APAL*, *GAL*, and *CAL*, the satisfiability problem is undecidable [9, 2]. The satisfiability problem of *AAML*, on the other hand, is decidable [10]. For *AAUL*, it remained unknown whether the satisfiability problem is decidable. Here, we show that it is *not* decidable, by demonstrating that *AAUL*'s satisfiability problem can encode the tiling problem [14]. Because the tiling problem is known to be co-RE complete [5], this shows that the satisfiability problem of *AAUL* is co-RE hard.

The undecidability result is not surprising, but also not obvious. In *APAL*, *GAL*, and *CAL* the undecidability seems to originate in the semantic restriction of quantification: the quantification is *only* over quantifier-free formulas, not over all formulas; the resulting gaps in the quantification make these logics more expressive than epistemic logic, and this also seems to affect decidability. However, in *AAML* it does not matter if we so restrict the semantics of quantifiers: either way, we can eliminate quantifiers from the language by rewriting procedures, and epistemic logic is decidable. As *AAUL* seems half-way between *APAL* and *AAML*, the scales could have tilted both towards decidability and undecidability.

The undecidability proof presented here is similar to those in [9] and [2] in that they all use the "arbitrary" operators to encode a grid and then reduce the tiling problem to a satisfiability problem on that grid. The similarities between the proofs do not go far beyond that, however.

The structure of this paper is as follows. First, in Section 2 we introduce the syntax and semantics of $AAUL$. Then, in Section 3 we provide a brief definition of the tiling problem and show that it can be encoded in the satisfiability problem of $AAUL$.

2. AAUL Syntax and Semantics

Let $\mathcal{P}$ be a countable set of propositional variables and $\mathcal{A}$ a finite set of agents. We assume that $|\mathcal{A}| \geq 6$.

Definition 1. The language $\mathcal{L}_{AAUL}$ of $AAUL$ is given by the following normal forms:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box_a \varphi \mid [U]\varphi \mid [\updownarrow]\varphi$$
$$U ::= (\varphi, a, \varphi) \mid U, (\varphi, a, \varphi)$$

where $p \in \mathcal{P}$ and $a \in \mathcal{A}$. The language $\mathcal{L}_{AUL}$ is the fragment of $\mathcal{L}_{AAUL}$ that does not contain $[\updownarrow]$.

We use $\vee, \rightarrow, \leftrightarrow, \Diamond, \langle U \rangle, \langle \updownarrow \rangle, \bigvee$ and $\bigwedge$ in the usual way as abbreviations. Furthermore, we slightly abuse notation by identifying the list $U = (\varphi_1, a_1, \psi_1), \cdots, (\varphi_k, a_k, \psi_k)$ with the set $U = \{(\varphi_1, a_1, \psi_1), \cdots, (\varphi_k, a_k, \psi_k)\}$. Finally, for $B \subseteq \mathcal{A}$ we use (φ, B, ψ) as an abbreviation for $\{(\varphi, a, \psi) \mid a \in B\}$.

$AAUL$ is evaluated on standard multi-agent Kripke models.

Definition 2. A model $\mathcal{M}$ is a triple $\mathcal{M} = (W, R, V)$ where W is a set of states, $R : \mathcal{A} \rightarrow 2^{W \times W}$ assigns to each agent an accessibility relation and $V : \mathcal{P} \rightarrow 2^W$ is a valuation.

Note that we are using the class of all Kripke models. This is unlike $APAL$ and GAL, which are typically considered on the class of S5 models.

Now, let us consider the semantics of $AAUL$. We start by giving the formal definition, after the definition we briefly discuss the intuition behind some of the operators.

Definition 3. Let $\mathcal{M} = (W, R, V)$ be a model and let $w \in W$. The satisfaction relation $\models$ is given by

$$\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & w \in V(p) \\
\mathcal{M}, w \models \neg\varphi & \text{iff} & \mathcal{M}, w \not\models \varphi \\
\mathcal{M}, w \models (\varphi \wedge \psi) & \text{iff} & \mathcal{M}, w \models \varphi \text{ and } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models \Box_a \varphi & \text{iff} & \mathcal{M}, v \models \varphi \text{ for each } v \text{ such that } (w, v) \in R(a) \\
\mathcal{M}, w \models [U]\varphi & \text{iff} & (\mathcal{M} * U), w \models \varphi \\
\mathcal{M}, w \models [\updownarrow]\varphi & \text{iff} & \mathcal{M}, w \models [U]\varphi \text{ for each } U \in L_{AUL}
\end{array}$$

where $(\mathcal{M} * U)$ is given by:

$$\begin{array}{lll}
\mathcal{M} * U & = & (W, R^U, V) \\
R^U(a) & = & \{(v, v') \in R(a) \mid \exists (\varphi, a, \varphi') \in U : \\
 & & \qquad (\mathcal{M}, v \models \varphi \text{ and } \mathcal{M}, v' \models \varphi')\}
\end{array}$$

A full discussion of the applications of $AAUL$ and of the intuitions behind the semantics of arrow updates and arbitrary arrow updates is outside the scope of this paper. For such a discussion, see [11] and [8]. However, in order to understand the undecidability proof it is important to grasp the semantics of $AAUL$. We therefore do provide a very brief explanation of the intuition behind and the semantics of $AAUL$.

Although our goal is to understand $AAUL$, is is useful to start by considering public announcements. We assume that the reader is familiar with public announcement logic, if not see for example [4]. A public announcement $[\psi]$ informs all agents that ψ is true. As a result, every possible world that the agents previously considered possible that does not satisfy ψ is rejected after the announcement, since it is incompatible with the new information. Semantically, this corresponds to a model $\mathcal{M}$ being transformed into a model $\mathcal{M} * \psi$ where all $\neg\psi$ states of $\mathcal{M}$ have been removed.

Like public announcements, arrow updates provide agents with new information. Unlike with public announcements, however, the new information provided by an arrow update can (i) differ per agent and (ii) differs per state. A typical example is a card game, where cards have been dealt face down. Now, agent a picks up her hand of cards and looks at it. Obviously, the information that a gains from this action is different than the information the other agents gain: a learns what her cards are whereas the other agents only learn that a now knows what her cards are. It is perhaps less obvious that the information that a gains also differs per state. Suppose that a has been dealt the 7 of Hearts. Then by looking at her cards a learns that she has the 7 of Hearts. If, on the other hand, a has been dealt the 8 of Clubs, then she learns that she has the 8 of Clubs. Learning that you have the 7 of Hearts is different from learning that you have the 8 of Clubs, so the information given to a depends on the state of the world.

With arrow updates we formalize the information that the agents gain in such a situation. In principle, we could do this in two ways: we could specify the things that are *incompatible* with the new information, or the things that are *compatible*. We choose to follow public announcements in this aspect, so just like $[\psi]$ says that the new information is compatible with ψ, we use an arrow update U to specify the information that is compatible with U. Since the information gained in an arrow update can depend on the agent and on the current state, we use triples (φ, a, ψ). We call such triples *clauses*; they can be read as "if the current state satisfies φ, then the information provided to agent a is compatible with ψ." Semantically, the effect of a triple (φ, a, ψ) is that every transition that is labeled a and that goes from a φ state to a ψ state is retained.

An arrow update is a finite set of clauses, $U = \{(\varphi_1, a_1, \psi_1), \cdots, (\varphi_k, a_k, \psi_k)\}$ (where it is possible that $\varphi_i = \varphi_j$, $a_i = a_j$ or $\psi_i = \psi_j$ for $i \neq j$). This still leaves the decision of what to do if a state matches multiple clauses. Suppose, for example, that $(\varphi_1, a, \psi_1), (\varphi_2, a, \psi_2) \in U$ and that a state satisfies both φ_1 and φ_2. There are several options for how to interpret this situation, we choose to interpret it disjunctively: if a state satisfies φ_1 and φ_2, then any state that satisfies ψ_1 *or* ψ_2 is consistent with the new information.

On the semantical level, this means that $\mathcal{M} * U$ should contain exactly those arrows of $\mathcal{M}$ that match at least one clause of U, where we say that $(w_1, w_2) \in R(a)$ matches (φ_1, a_1, ψ_1) if and only if $\mathcal{M}, w_1 \models \varphi_1$, $a = a_1$ and $\mathcal{M}, w_2 \models \psi_1$.

Arbitrary arrow updates then quantify over such arrow updates. However, in order to avoid circularity we restrict this quantification to those arrow updates that do not themselves contain an arbitrary arrow update $[\updownarrow]$. So $\mathcal{M}, w \models [\updownarrow]\varphi$ if and only if $\mathcal{M}, w \models [U]\varphi$ for all $U \in \mathcal{L}_{AUL}$.

3. Reducing the Tiling Problem

3.1. The Tiling Problem

We will prove the undecidability of *AAUL* by a reduction of the tiling problem. The tiling problem was introduced in [14] and can be defined as follows.

Definition 4. Let C be a finite set of colors. A *tile type* is a function $i : \{north, south, east, west\} \to C$.

An instance of the tiling problem is a finite set *types* of tile types. A solution to an instance of the tiling problem is a function $tiling : \mathbb{Z} \times \mathbb{Z} \to types$ such that, for every $(z_1, z_2) \in \mathbb{Z} \times \mathbb{Z}$,

$$\begin{aligned} tiling(z_1, z_2)(north) &= tiling(z_1, z_2 + 1)(south) \\ tiling(z_1, z_2)(east) &= tiling(z_1 + 1, z_2)(west). \end{aligned}$$

The tiling problem was shown to be undecidable in [5]. In fact, the tiling problem is co-RE complete. Therefore, by reducing the tiling problem to the satisfiability problem of *AAUL*, we show that the latter problem is co-RE hard. Whether *AAUL*'s satisfiability problem is co-RE is not currently known.

3.2. Encoding the Tiling Problem in AAUL

We want to encode the tiling problem in *AAUL*. So for every instance *types* of the tiling problem we define a formula χ_{types} of *AAUL* that is satisfiable if and only if *types* can tile the plane. The strategy for doing this is as follows.

We represent each point of $\mathbb{Z} \times \mathbb{Z}$ by a state (n, m). For every $i \in types$ we then use a propositional variable p_i to represent "the current state contains a tile of type i." For every $c \in C$ we use propositional variables $north_c$ (resp. $south_c$, $east_c$, $west_c$) to represent the northern (resp. southern, eastern, western) edge of the current tile having color c. Finally, we use relations *up*, *down*, *left* and *right* to represent one tile being above, below, to the left and to the right, respectively, of the current tile.

In addition to the states (n, m) that correspond to points in $\mathbb{Z} \times \mathbb{Z}$, we also use an auxiliary state s_0. This state s_0 is not part of the grid, and does not contain any tile. Instead, it is the state where χ_{types} will be evaluated. We therefore also refer to s_0 as the origin state. In order to distinguish s_0 from the states that are part of the grid we use the propositional variable p, which holds on s_0 but not on any (n, m).

Now, given any state (n, m), it is relatively easy to check whether the constraints of a tiling are satisfied locally. For example, $\bigvee_{i \in types} p_i \wedge \bigwedge_{i \neq j \in types} \neg(p_i \wedge p_j)$ holds if and only if the current state has exactly one type of tile, and $\bigwedge_{c \in C}(north_c \to \Box_{up} south_c)$ holds if and only if the northern color of the current tile matches the southern color of the tile above.

Making sure that the *global* constraints of a tiling are satisfied is harder, though. We do this in the following way. Firstly, we take a relation $R(b)$, and force it to connect between the auxiliary state s_0 and every state (n, m).[2] So while $\bigvee_{i \in types} p_i \wedge \bigwedge_{i \neq j \in types} \neg(p_i \wedge p_j)$ says that the current state has exactly one tile type, the formula $\Box_b \bigvee_{i \in types} p_i \wedge \bigwedge_{i \neq j \in types} \neg(p_i \wedge p_j)$ says that *all* grid states have exactly one tile type. Secondly, we enforce a grid-like structure onto the domain.

We also use another relation $R(a)$ in order to simulate a Boolean variable: every state will have an a-arrow to itself (or at least, to a modally indistinguishable state). If an arrow update retains the a-arrow departing from a state s we can see this as the variable being true on s, and if an arrow update removes the a-arrow departing from s we can see this as the variable being false on s.

With the above in mind, let us define the formula χ_{types}.

Definition 5. Let *types* be an instance of the tiling problem. The formula χ_{types} is given by

$$\chi_{types} := \psi_{grid} \wedge [U_{grid}]\psi_{grid} \wedge \psi_{types}$$

where

$$\begin{aligned}
\psi_{grid} &:= \psi_1 \wedge \bigwedge_{x \in D} (\psi_{2,x} \wedge \psi_{3,x} \wedge propd_x \wedge return_x) \wedge inverse \wedge commute \\
U_{grid} &:= (p \to \psi_{grid}, a, \top), (\top, \mathcal{A} \setminus \{a\}, \top) \\
\psi_{types} &:= one_tile \wedge one_color \wedge tile_colors \wedge tile_match
\end{aligned}$$

and

$$\begin{aligned}
D &:= \{up, down, left, right\} \\
\psi_1 &:= \Diamond_a \top \wedge [\updownarrow](\Diamond_a \top \to \Box_b \Box_b \Diamond_a \top) \\
\psi_{2,x} &:= p \wedge \Diamond_b \top \wedge \Box_b(\neg p \wedge \Diamond_b p \wedge \Diamond_x(\neg p \wedge \Diamond_b p) \wedge \Box_x \neg p) \\
\psi_{3,x} &:= \Box_b(ref \wedge \Box_x ref \wedge [\updownarrow](\Diamond_x \Diamond_a \top \to \Box_x \Diamond_a \top)) \\
ref &:= \Diamond_a \top \wedge [\updownarrow] \Box_a \Diamond_a \top \\
propd_x &:= \Box_b [\updownarrow]((\Box_a \bot \wedge \Diamond_x \Diamond_a \top \wedge \Diamond_b(\Diamond_b \top \wedge \Box_b \Diamond_a \top) \wedge \\
&\qquad \langle\updownarrow\rangle(\Diamond_x \Diamond_a \top \wedge \Diamond_b \Diamond_b \Box_a \bot)) \to [U_x]\langle\updownarrow\rangle(\Diamond_x \Diamond_a \top \wedge \Diamond_b \Diamond_b \Box_a \bot)) \\
U_x &:= (p \vee \Box_a \bot, b, \top), (\top, a, \top), (\Box_a \bot, x, \top)
\end{aligned}$$

[2]This is far easier said then done, we will spend several pages proving that $R(b)$ connects to every relevant state.

$$\begin{aligned}
\mathit{return}_x &:= \Box_b\langle\updownarrow\rangle(\Box_a\bot \wedge \Diamond_x\Diamond_a\top \wedge \Diamond_b(\Diamond_b\top \wedge \Box_b\Diamond_a\top)\wedge\\
&\ [\updownarrow](\Diamond_x\Diamond_a\top \to \Box_b\Box_b\Diamond_a\top))\\
\mathit{inverse} &:= \Box_b[\updownarrow](\Box_a\bot \to (\Box_{up}\Box_{down}\Box_a\bot \wedge \Box_{down}\Box_{up}\Box_a\bot\wedge\\
&\ \Box_{left}\Box_{right}\Box_a\bot \wedge \Box_{right}\Box_{left}\Box_a\bot))\\
\mathit{commute} &:= \Box_b[\updownarrow]\bigwedge_{(x,y)\in E}(\Diamond_x\Diamond_y\Box_a\bot \to \Box_y\Box_x\Box_a\bot)\\
E &:= \{(up, left), (up, right), (down, left), (down, right),\\
&\ (left, up), (left, down), (right, up), (right, down)\}\\
\mathit{one_tile} &:= \Box_b(\bigvee_{i\in tiles} p_i \wedge \bigwedge_{i\neq j\in tiles}\neg(p_i \wedge p_j))\\
\mathit{one_color} &:= \Box_b\bigwedge_{c\in C}(north_c \to \bigwedge_{d\in C\setminus\{c\}}\neg north_d)\wedge\\
&\ \Box_b\bigwedge_{c\in C}(south_c \to \bigwedge_{d\in C\setminus\{c\}}\neg south_d)\wedge\\
&\ \Box_b\bigwedge_{c\in C}(east_c \to \bigwedge_{d\in C\setminus\{c\}}\neg east_d)\wedge\\
&\ \Box_b\bigwedge_{c\in C}(west_c \to \bigwedge_{d\in C\setminus\{c\}}\neg west_d)\\
\mathit{tile_colors} &:= \Box_b\bigwedge_{i\in tiles}(p_i \to (north_{i(north)}\wedge\\
&\ south_{i(south)} \wedge east_{i(east)} \wedge west_{i(west)}))\\
\mathit{tile_match} &:= \Box_b\bigwedge_{c\in C}((north_c \to \Box_{up} south_c) \wedge (west_c \to \Box_{left} east_c))
\end{aligned}$$

Note that the formulas $\psi_{2,x}$, $\psi_{3,x}$, $propd_x$ and $return_x$ and the update U_x contain a parameter x, which ranges over the four directions $D = \{up, down, left, right\}$.

The formula ψ_{grid}, together with $[U_{grid}]\psi_{grid}$, encodes a grid. The formula ψ_{types} then ensures that the grid is tiled with tiles from *types*. The formula χ_{grid} may look rather intimidating, but we will discuss the various subformulas in detail and explain what they do.

We want to show that χ_{types} is satisfiable if and only if *types* can tile $\mathbb{Z}\times\mathbb{Z}$. We start by showing that if such a tiling exists, then χ_{types} is satisfiable.

Lemma 1. *Suppose types can tile* $\mathbb{Z}\times\mathbb{Z}$. *Then* χ_{types} *is satisfiable.*

Proof. Let *tiling* be the tiling, let $p_{n,m} \in \mathcal{P}$ for every $n, m \in \mathbb{Z}$ and let $\mathcal{M} = (S, R, V)$ be the following, quite straightforward, encoding of *tiling*:

- $S = (\mathbb{Z}\times\mathbb{Z}) \cup s_0$
- $R(a) = \{(s, s) \mid s \in S\}$

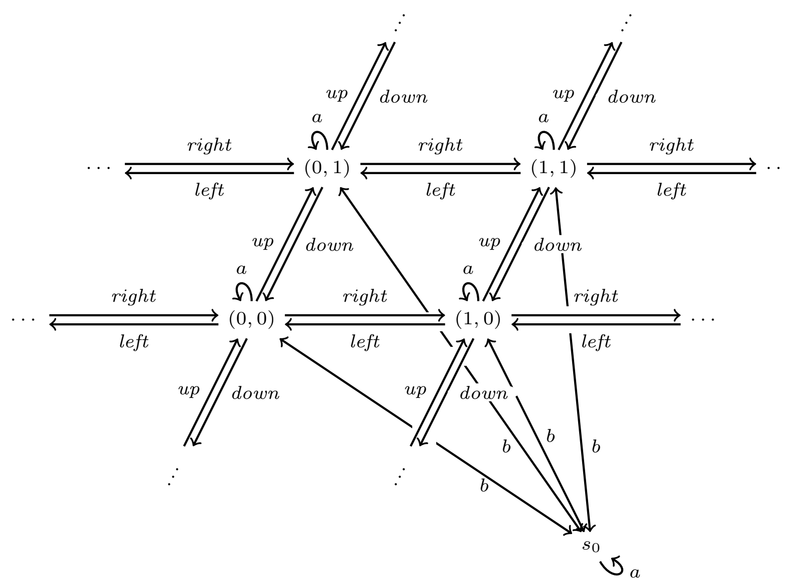


Figure 1: The model used in Lemma 1.

- $R(b) = \{(s_0, (n, m)) \mid n, m \in \mathbb{Z}\} \cup \{((n, m), s_0) \mid n, m \in \mathbb{Z}\}$
- $R(up) = \{((n, m), (n, m + 1)) \mid n, m \in \mathbb{Z}\}$
- $R(down) = \{((n, m), (n, m - 1)) \mid n, m \in \mathbb{Z}\}$
- $R(left) = \{((n, m), (n - 1, m)) \mid n, m \in \mathbb{Z}\}$
- $R(right) = \{((n, m), (n + 1, m)) \mid n, m \in \mathbb{Z}\}$
- $V(p) = \{s_0\}$
- $V(p_i) = \{(n, m) \mid tiling(n, m) = i\}$ for $i \in tiles$
- $V(north_c) = \{(n, m) \mid ((tiling)(n, m))(north) = c\}$ for $c \in C$
- $V(south_c) = \{(n, m) \mid ((tiling)(n, m))(south) = c\}$ for $c \in C$
- $V(east_c) = \{(n, m) \mid ((tiling)(n, m))(east) = c\}$ for $c \in C$
- $V(west_c) = \{(n, m) \mid ((tiling)(n, m))(west) = c\}$ for $c \in C$
- $V(p_{n,m}) = \{(n, m)\}$

The frame of this model (i.e. the model without the valuation) is also drawn in Figure 1.

As mentioned above, the state s_0 is special: it is the origin state, and the only state that does not have a tile type associated with it. The propositional variable p is used to identify this special state. First, we will show that $\mathcal{M}, s_0 \models \psi_{grid}$.

There is an a-arrow from s_0 to itself, so $\mathcal{M}, s_0 \models \Diamond_a \top$. Furthermore, the only b-b-successor of s_0 is s_0 itself. It follows that every arrow update that retains the a-arrow from s_0 also retains the a-arrow from every b-b-successor of s_0. So $\mathcal{M}, s_0 \models [\updownarrow](\Diamond_a \top \rightarrow \Box_b \Box_b \Diamond_a \top)$. We have shown that s_0 satisfies both conjuncts of ψ_1, so $\mathcal{M}, s_0 \models \psi_1$.

The state s_0 satisfies p and it has at least one b-successor, so $\mathcal{M}, s_0 \models p \wedge \Diamond_b \top$. Every state (n, m) satisfies $\neg p$ and has a b-arrow to the p-state s_0. Furthermore, for every $x \in D$ the state (n, m) has exactly one x-successor (n', m'), that also satisfies $\neg p \wedge \Diamond_b p$. Since every b-successor of s_0 is a state (n, m), it follows that s_0 satisfies $p \wedge \Diamond_b \top \wedge \Box_b(\neg p \wedge \Diamond_b p \wedge \Diamond_x(\neg p \wedge \Diamond_b p) \wedge \Box_x \neg p)$ for every $x, \in D$, so $\mathcal{M}, s_0 \models \psi_{2,x}$.

Now, consider the formula *ref*. Every state s of $\mathcal{M}$ has exactly one outgoing a-arrow, and that a-arrow goes to s itself. It is therefore impossible to have an arrow update that retains the a-arrow from s to one of its a-successors s' while removing all a-arrows from s'. It follows that every state of $\mathcal{M}$ satisfies $\Diamond_a \top \wedge [\updownarrow] \Box_a \Diamond_a \top$, so all states satisfy *ref*.

Now, take any direction $x \in D$. From the fact that every state of $\mathcal{M}$ satisfies *ref*, it follows that every state (n, m) satisfies $ref \wedge \Box_x ref$. Furthermore, every state (n, m) has exactly one x-successor, so every arrow update that retains the a-arrow on one of the x-successors of (n, m) retains the arrow on every x-successor of (n, m). In other words, we have $\mathcal{M}, (n, m) \models [\updownarrow](\Diamond_x \Diamond_a \top \rightarrow \Box_x \Diamond_a \top)$. Together with the fact that (n, m) satisfies $ref \wedge \Box_x ref$, as discussed earlier, this implies that $\mathcal{M}, (n, m) \models (ref \wedge \Box_x ref \wedge [\updownarrow](\Diamond_x \Diamond_a \top \rightarrow \Box_x \Diamond_a \top))$. The above holds for every state (n, m) and every $x \in D$, so $\mathcal{M}, s_0 \models \psi_{3,x}$ for every $x \in D$.

Let us then consider $propd_x$. For ease of notation we show only that $propd_{right}$ holds; the other directions can be proven in the same way. The initial $\Box_b$ operator of $propd_{right}$ takes us to any state (n, m). To show is that

$$\mathcal{M}, (n, m) \models [\updownarrow]((\Box_a \bot \wedge \Diamond_{right} \Diamond_a \top \wedge \Diamond_b(\Diamond_b \top \wedge \Box_b \Diamond_a \top) \wedge \langle \updownarrow \rangle (\Diamond_{right} \Diamond_a \top \wedge \Diamond_b \Diamond_b \Box_a \bot)) \rightarrow [U_{right}] \langle \updownarrow \rangle (\Diamond_{right} \Diamond_a \top \wedge \Diamond_b \Diamond_b \Box_a \bot))$$

Let U_1 be any arrow update such that the antecedent in the above formula is true, i.e. any arrow update such that

$$\mathcal{M} * U_1, (n, m) \models (\Box_a \bot \wedge \Diamond_{right} \Diamond_a \top \wedge \Diamond_b(\Diamond_b \top \wedge \Box_b \Diamond_a \top) \wedge \langle \updownarrow \rangle (\Diamond_{right} \Diamond_a \top \wedge \Diamond_b \Diamond_b \Box_a \bot).$$

By $\Box_a \bot$ the a-arrow on (n, m) was removed by U_1. By $\Diamond_{right} \Diamond_a \top$ the *right*-arrow to $(n+1, m)$ and the a-arrow on $(n+1, m)$ are retained. By $\Diamond_b(\Diamond_b \top \wedge \Box_b \Diamond_a \top)$, the arrow from (n, m) to s_0 is retained, as well as a b-arrow from s_0 to at least one state (n', m'). Furthermore, every b-arrow from s_0 that is retained, points to a state that still has its a-arrow.

The formula $\langle \updownarrow \rangle (\Diamond_{right} \Diamond_a \top \wedge \Diamond_b \Diamond_b \Box_a \bot)$ then states that there is some update U_2 such that $(\mathcal{M} * U_1) * U_2, (n, m) \models \Diamond_{right} \Diamond_a \top \wedge \Diamond_b \Diamond_b \Box_a \bot$. Note that this is impossible if $(n+1, m)$ is the only b-b-successor of (n, m) in $\mathcal{M} * U_1$, since then

$(n+1,m)$ would need to satisfy $\Diamond_a\top$ (due to (n,m) satisfying $\Diamond_{right}\Diamond_a\top$) as well as $\Box_a\bot$ (due to (n,m) satisfying $\Diamond_b\Diamond_b\Box_a\bot$) in $(\mathcal{M}*U_1)*U_2$.

To show is that for every such U_1, we have

$$\mathcal{M}*U_1,(n,m)\models[U_{right}]\langle\updownarrow\rangle(\Diamond_{right}\Diamond_a\top\wedge\Diamond_b\Diamond_b\Box_a\bot),$$

where $U_{right}=(p\vee\Box_a\bot,b,\top),(\top,a,\top),(\Box_a\bot,right,\top)$. Note that U_{right} retains all a-arrows, the *right*-arrow from (n,m) to $(n+1,m)$ and the b-arrow from (n,m) to s_0 (because $\mathcal{M}*U_1,(n,m)\models\Box_a\bot$) as well as all b-arrows from s_0 (because $\mathcal{M}*U_1,s_0\models p$).

Now, let $U_2:=(p_{n+1,m},a,\top),(\top,\{b,right\},\top)$. This update retains all b- and *right*-arrows as well as the a-arrow on $(n+1,m)$ while removing all other a-arrows. Since (n,m) had at least one b-b-successor $(n',m')\neq(n,m)$ in $(\mathcal{M}*U_1)*U_{right}$, it follows that, in $((\mathcal{M}*U_1)*U_{right})*U_2$the state s_0 has a *right*-successor that satisfies $\Diamond_a\top$ (namely (n,m)) and a b-b-successor that satisfies $\Box_a\bot$ (namely (n',m')). We therefore have $((\mathcal{M}*U_1)*U_{right})*U_2,(n,m)\models\Diamond_{right}\Diamond_a\top\wedge\Diamond_b\Diamond_b\Box_a\bot$ and therefore $\mathcal{M}*U_1,(n,m)\models[U_{right}]\langle\updownarrow\rangle(\Diamond_{right}\Diamond_a\top\wedge\Diamond_b\Diamond_b\Box_a\bot)$. We have now shown that $\mathcal{M},s_0\models propd_{right}$.

We continue with $return_x$. Once again, we consider the case $x=right$, the other directions can be proven in a similar way. The formula $return_{right}$ starts with a $\Box_b$ operator, so take any b-successor (n,m) of s_0. Furthermore, let $U_1:=(p_{n+1,m},a,\top),(\top,b,p\vee p_{n+1,m}),(\top,right,\top)$. So in $\mathcal{M}*U_1$ the state $(n+1,m)$ is the only one to still have its a-arrow, and it is also the only *right*- and b-b-successor of (n,m). Hence $\mathcal{M}*U_1,(n,m)\models\Box_a\bot\wedge\Diamond_{right}\Diamond_a\top\wedge\Diamond_b(\Diamond_b\top\wedge\Box_b\Diamond_a\top)$. Furthermore, any arrow update that removes the a-arrow from the *right*-successor of (n,m) also removes the a-arrow from the b-b-successor of (n,m), since those successors are the same state $(n+1,m)$. Therefore, $\mathcal{M}*U_1,(n,m)\models[\updownarrow](\Diamond_{right}\Diamond_a\top\rightarrow\Box_b\Box_b\Diamond_a\top)$. Putting these things together, we obtain

$$\begin{aligned}\mathcal{M},(n,m)\models[U_1](&\Box_a\bot\wedge\Diamond_{right}\Diamond_a\top\wedge\Diamond_b(\Diamond_b\top\wedge\Box_b\Diamond_a\top)\wedge\\&[\updownarrow](\Diamond_{right}\Diamond_a\top\rightarrow\Box_b\Box_b\Diamond_a\top))\end{aligned}$$

and therefore

$$\begin{aligned}\mathcal{M},(n,m)\models\langle\updownarrow\rangle(&\Box_a\bot\wedge\Diamond_{right}\Diamond_a\top\wedge\Diamond_b(\Diamond_b\top\wedge\Box_b\Diamond_a\top)\wedge\\&[\updownarrow](\Diamond_{right}\Diamond_a\top\rightarrow\Box_b\Box_b\Diamond_a\top)).\end{aligned}$$

Since this holds for any b-successor (n,m) of s_0, it follows that $return_{right}$ holds in s_0.

We continue with *inverse*. In $\mathcal{M}$, the relations *up* and *down* are each others inverses, as are *left* and *right*. Furthermore, all four direction relations are functions. It follows immediately that, for every (n,m), we have $\mathcal{M},(n,m)\models[\updownarrow](\Box_a\bot\rightarrow\Box_{right}\Box_{left}\Box_a\bot)$, and similarly for the other combinations of directions. So we have $\mathcal{M},s_0\models inverse$.

Similarly, in $\mathcal{M}$ we have $R(right)\circ R(up)=R(up)\circ R(right)$, and the same for the other directions. It follows that $\mathcal{M},s_0\models\Box_b[\updownarrow]\bigwedge_{(x,y)\in E}(\Diamond_x\Diamond_y\Box_a\bot\rightarrow\Box_y\Box_x\Box_a\bot)$.

We have now considered all the conjuncts of ψ_{grid}, so we have shown that $\mathcal{M}, s_0 \models \psi_{grid}$. Furthermore, the only p state in $\mathcal{M}$ is the state s_0, and s_0 satisfies ψ_{grid}. The update $U_{grid} = (p \rightarrow \psi_{grid}, a, \top), (\top, \mathcal{A} \setminus \{a\}, \top)$ therefore retains all arrows. So ψ_{grid} remains true after this update, which gives us $\mathcal{M}, s_0 \models [U_{grid}]\psi_{grid}$.

This only leaves the formula ψ_{types}. This formula simply encodes that *tiling* is a tiling on $\mathbb{Z} \times \mathbb{Z}$, so it is straightforward to verify that $\mathcal{M}, s_0 \models \psi_{types}$.

We have now shown that all the conjuncts of χ_{types} are satisfied in $\mathcal{M}, s_0$, so $\mathcal{M}, s_0 \models \chi_{types}$, which was to be shown. $\square$

We have shown that if *types* can tile the plane, then χ_{types} is satisfiable. Left to show is that if χ_{types} is satisfiable, then *types* can tile the plane. The main strategy that we use in this proof is to show that the subformulas ψ_{grid} and $[U_{grid}]\psi_{grid}$ of χ_{types} only hold in models that resemble the grid-like model shown in Figure 1. The subformula ψ_{types} of χ_{types} then only holds if the grid can be tiled with *types*.

Unfortunately, there is one significant complication. The language of $AAUL$ is not expressive enough to guarantee uniqueness of states. So, for example, a state (n, m) may have two (or more) different *right*-successors, $(n+1, m)$ and $(n+1, m)'$. We can, however, use ψ_{grid} to show that if $(n+1, m)$ and $(n+1, m)'$ are both *right*-successors of (n, m), then $(n+1, m)$ and $(n+1, m)'$ are modally indistinguishable. So a pointed model where χ_{types} is satisfied resembles the model from Figure 1 modulo modal indistinguishability. This suffices to show that χ_{types} is only satisfiable if *types* can tile $\mathbb{Z} \times \mathbb{Z}$.

In Lemma 5 we will prove that satisfiability of χ_{types} implies that *types* can tile the plane. Before doing so, however, it is useful to consider a few auxiliary definitions and lemmas.

Definition 6. Fix a state s_0, and let $\mathcal{M} = (S, R, V)$ be any model that has s_0 as one of its states. The set $[s_0]_{\mathcal{M}}$ is the smallest set of states of $\mathcal{M}$ such that

- $s_0 \in [s_0]_{\mathcal{M}}$ and
- if $s \in [s_0]_{\mathcal{M}}$ and $(s, s') \in R_b \circ R_b$ then $s' \in [s_0]_{\mathcal{M}}$.

Lemma 2. *Suppose $\mathcal{M}, s_0 \models \psi_{grid}$, and let s be any b-b-successor of s_0. Then s_0 and s are modally indistinguishable.*

Proof. Suppose towards a contradiction that there is a modal formula δ such that $\mathcal{M}, s_0 \models \delta$ and $\mathcal{M}, s \not\models \delta$.

From $\mathcal{M}, s_0 \models \psi_{grid}$ it follows that, in particular, $\mathcal{M}, s_0 \models \psi_1$ and therefore (by definition) $\mathcal{M}, s_0 \models \Diamond_a \top \wedge [\updownarrow](\Diamond_a \top \rightarrow \Box_b \Box_b \Diamond_a \top)$. The $\Diamond_a \top$ subformula implies that s_0 has at least one a-successor s_0'.

Consider the update $U = (\delta, a, \top), (\top, b, \top)$. This U retains all b-arrows, so s is still a b-b-successor of s_0 in the updated model $\mathcal{M} * U$. Furthermore, since U retains exactly those a-arrows that depart from a δ-world, we have $\mathcal{M} * U, s_0 \models \Diamond_a \top$ and $\mathcal{M} * U, s \not\models \Diamond_a \top$. It follows that $\mathcal{M}, s_0 \models \neg[U](\Diamond_a \top \rightarrow \Box_b \Box_b \Diamond_a \top)$, contradicting $\mathcal{M}, s_0 \models [\updownarrow](\Diamond_a \top \rightarrow \Box_b \Box_b \Diamond_a \top)$.

Our assumption that a distinguishing modal formula δ exists must therefore have been false, so s_0 and s are modally indistinguishable. $\square$

Lemma 3. *If $\mathcal{M}, s_0 \models \psi_{grid}$, then all elements of $[s_0]_\mathcal{M}$ are modally indistinguishable from s_0.*

Proof. Let $\mathcal{M}, s_0$ be any pointed model such that $\mathcal{M}, s_0 \models \psi_{grid}$, and let s be any element of $[s_0]_\mathcal{M}$. Then there is a sequence $s_0, s_1, \cdots, s_n$ of states such that $s = s_n$ and, for every $0 \leq i < n$, the state s_{i+1} is a b-b-successor of s_i.

We show that s is modally indistinguishable from s_0, by induction on n. As base case, suppose $n = 1$. Then it follows immediately from Lemma 2 that s and s_0 are modally indistinguishable. Assume then as induction hypothesis that $n > 1$ and that $s_0, s_1, \cdots, s_{n-1}$ are modally indistinguishable from s_0.

If s is modally distinguishable from s_0, then there is some modal formula δ that holds on s_0 but not on s. Since s_0 is modally indistinguishable from its b-b-successors, we also have $\mathcal{M}, s_0 \models \Box_b\Box_b\delta$. However, since s is a b-b-successor of s_{n-1} and δ does not hold on s, we have $\mathcal{M}, s_{n-1} \not\models \Box_b\Box_b\delta$. This implies that there is a modal formula that distinguishes between s_0 and s_{n-1}, contradicting the induction hypothesis.

It follows that there can be no modal δ that holds on s_0 but not on s. This completes the induction step and thereby the proof. $\square$

Lemma 4. *If $\mathcal{M}, s_0 \models \psi_{grid} \land [U_{grid}]\psi_{grid}$, then all elements of $[s_0]_\mathcal{M}$ satisfy ψ_{grid}.*

Proof. First, note that $\mathcal{M}, s_0 \models p$, because $\mathcal{M}, s_0 \models \psi_{grid}$ and therefore $\mathcal{M}, s_0 \models \psi_{2,x}$. By Lemma 3, all elements of $[s_0]_\mathcal{M}$ are modally indistinguishable, so all of them satisfy p.

Now, take any $s \in [s_0]_\mathcal{M}$. Then there is a finite sequence $s_0, s_1, \cdots, s_n$ of states such that $s = s_n$ and for every $0 \leq i < n$ the state s_{i+1} is a, b-b-successor of s_i.

Recall that $U_{grid} = (p \to \psi_{grid}, a, \top), (\top, \mathcal{A} \setminus \{a\}, \top)$. The b-arrows on the path from s_0 to s are retained by U_{grid} since that update retains all b-arrows. This implies that $s \in [s_0]_{\mathcal{M} * U_{grid}}$. Furthermore, $\mathcal{M} * U_{grid}, s_0 \models \psi_{grid}$ because, by the assumptions of the lemma, $\mathcal{M}, s_0 \models [U_{grid}]\psi_{grid}$. Lemma 3 therefore implies that s_0 and s are modally indistinguishable in $\mathcal{M} * U_{grid}$.

By the assumptions of the lemma, $\mathcal{M}, s_0 \models \psi_{grid}$. This implies that s_0 has at least one a-successor in $\mathcal{M}$, and that the a-arrow to this successor is retained by the update. So $\mathcal{M} * U_{grid}, s_0 \models \Diamond_a\top$.

Suppose towards a contradiction that $\mathcal{M}, s \not\models \psi_{grid}$. Then $\mathcal{M}, s \not\models p \to \psi_{grid}$, so the update U_{grid} would remove all a-arrows from s and we would have $\mathcal{M} * U_{grid}, s \not\models \Diamond_a\top$. This would contradict the modal indistinguishability of s_0 and s in $\mathcal{M} * U_{grid}$. It follows that $\mathcal{M}, s \models \psi_{grid}$, which is what was to be shown. $\square$

Having dealt with these preliminaries, we can show that satisfiability of χ_{types} implies that *types* can tile $\mathbb{Z} \times \mathbb{Z}$.

Lemma 5. *If χ_{types} is satisfiable, then types can tile $\mathbb{Z} \times \mathbb{Z}$.*

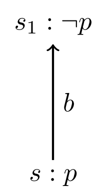
Proof. Let $\mathcal{M}, s_0$ be any pointed model such that $\mathcal{M}, s_0 \models \chi_{types}$. Then, in particular, $\mathcal{M}, s_0 \models \psi_{grid} \wedge [U_{grid}]\psi_{grid}$ and therefore all elements of $[s_0]_{\mathcal{M}}$ satisfy ψ_{grid} and are modally indistinguishable from each other.

We will now explain that the fact that ψ_{grid} holds on all of $[s_0]_{\mathcal{M}}$ implies that $\mathcal{M}$ is "grid-like." During this explanation, it is useful to draw diagrams of the model $\mathcal{M}$. Because $\mathcal{M}$ may be infinitely large, it is not very practical to draw the entire model, so we will only draw the parts that are relevant to the part of the proof they are intended to illustrate.

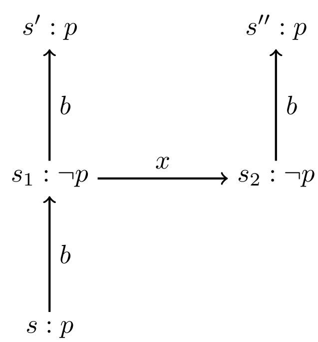
Take any $s \in [s_0]_{\mathcal{M}}$. We start by considering the formula $\psi_{2,x}$, that holds in s for every $x \in D$. So, by the definition of $\psi_{2,x}$, we have

$$\mathcal{M}, s \models p \wedge \Diamond_b \top \wedge \Box_b(\neg p \wedge \Diamond_b p \wedge \Diamond_x(\neg p \wedge \Diamond_b p) \wedge \Box_x \neg p).$$

This implies that s satisfies p, that s has at least one b-successor and that every b-successor s_1 of s satisfies $\neg p$. So far, this can be drawn as follows.



Furthermore, this s_1 has at least one b-successor s' that satisfies p, and one x-successor s_2 that satisfies $\neg p \wedge \Diamond_b p$. Our diagram becomes the following.



Now, consider $\psi_{3,x}$, which is also one of the conjuncts of ψ_{grid} and therefore holds on s. It states that

$$\mathcal{M}, s \models \Box_b(\mathit{ref} \wedge \Box_x \mathit{ref} \wedge [\updownarrow](\Diamond_x \Diamond_a \top \to \Box_x \Diamond_a \top))$$

So s_1 and s_2 both satisfy ref. This implies that s_1 and s_2 both have at least one outgoing a-arrow. Suppose now, towards a contradiction, that s_1 has an a-successor s_1' that is modally distinguishable from s_1. Then there is some modal formula δ that holds on s_1 but not on s_1', so we would have $\mathcal{M}, s_1 \models$

$[(\delta, a, \top)]\neg\Box_a\Diamond_a\top$, contradicting the $[\updownarrow]\Box_a\Diamond_a\top$ part of *ref*. It follows that s_1 is modally indistinguishable from all its a-successors. The same reasoning shows that s_2 is also modally indistinguishable from its a-successors.

Additionally, s_1 satisfies $[\updownarrow](\Diamond_x\Diamond_a\top \rightarrow \Box_x\Diamond_a\top)$. This implies that all x-successors of s_1 are modally indistinguishable from one another, since otherwise there would be some arrow update that retains the a-arrow from s_2 while removing all a-arrows from at least one of the other x-successors of s_1.

Now we should consider the two most complex conjuncts of ψ_{grid}: $propd_x$ and $return_x$. The formula $propd_x$ states that

$$\begin{aligned}\mathcal{M}, s \models \Box_b[\updownarrow]((\Box_a\bot \wedge \Diamond_x\Diamond_a\top \wedge \Diamond_b(\Diamond_b\top \wedge \Box_b\Diamond_a\top)\wedge \\ \langle\updownarrow\rangle(\Diamond_x\Diamond_a\top \wedge \Diamond_b\Diamond_b\Box_a\bot)) \rightarrow [U_x]\langle\updownarrow\rangle(\Diamond_x\Diamond_a\top \wedge \Diamond_b\Diamond_b\Box_a\bot)),\end{aligned}$$

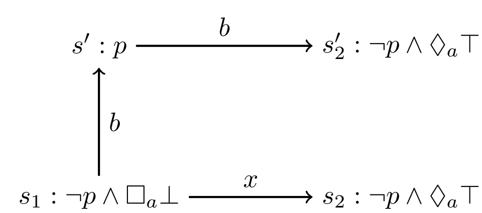
where

$$U_x = (p \vee \Box_a\bot, b, \top), (\top, a, \top), (\Box_a\bot, x, \top)$$

The initial $\Box_b$ of $propd_x$ takes us to any b-successor of s, so to s_1 in the above diagram. The remainder of $propd_x$ now has the form $[\updownarrow](\varphi_1 \rightarrow \varphi_2)$. So for any arrow update $U_1 \in \mathcal{L}_{AUL}$, if the antecedent φ_1 holds in $\mathcal{M} * U_1, s_1$ then the consequent φ_2 should hold there as well. Let us take a closer look at what it means for φ_1 to hold in $\mathcal{M} * U_1, s_1$.

We have $\varphi_1 = (\Box_a\bot \wedge \Diamond_x\Diamond_a\top \wedge \Diamond_b(\Diamond_b\top \wedge \Box_b\Diamond_a\top) \wedge \langle\updownarrow\rangle(\Diamond_x\Diamond_a\top \wedge \Diamond_b\Diamond_b\Box_a\bot)$. So s_1 satisfies $\Box_a\bot$. Furthermore, at least one x-successor of s_1 satisfies $\Diamond_a\top$. Since the x-successors of s_1 are modally indistinguishable (in $\mathcal{M}$, and therefore also in $\mathcal{M} * U_1$) this implies that all x-successors of s_1 retain their a-arrow.

The b-successor s' of s_1 (which is also unique up to modal indistinguishability) satisfies $\Diamond_b\top \wedge \Box_b\Diamond_a\top$. So the situation we are in can be represented by the diagram drawn below.



Note that s'_2 is a b-successor of a state $s' \in [s_0]_{\mathcal{M}}$. Above we concluded that any b-successor s_1 of any state $s \in [s_0]_{\mathcal{M}}$ must be modally indistinguishable from all its a-successors, so s'_2 is also modally indistinguishable from its a-successors.

The final conjunct of the antecedent φ_1 now states that there is some arrow update U_2 that retains the a-arrow from s_2, while removing the a-arrow from s'_2. This is the case if and only if s_2 and s'_2 are modally distinguishable.

The consequent φ_2 then states that $\langle\updownarrow\rangle(\Diamond_x\Diamond_a\top \wedge \Diamond_b\Diamond_b\Box_a\bot)$ still holds after the application of U_x, so $(\mathcal{M} * U_1) * U_x, s_1 \models \langle\updownarrow\rangle(\Diamond_x\Diamond_a\top \wedge \Diamond_b\Diamond_b\Box_a\bot)$. This implies that, in $(\mathcal{M} * U_1) * U_x$, the states s_2 and s'_2 are still modally distinguishable.

The update U_x removes all $\mathcal{A}\setminus\{a\}$-arrows from s_2 and s_2'. Furthermore, the a-arrows go from s_2 to a state modally indistinguishable from s_2 and from s_2' to a state modally indistinguishable from s_2'. This implies that, after the update U_x, the states s_2 and s_2' can only be modally distinguishable from each other if they are propositionally distinguishable.

In summary: $propd_x$ guarantees that if $\mathcal{M} * U_1$ matches the above diagram (i.e. $\mathcal{M} * U_1, s_1 \models \Box_a\bot \wedge \Diamond_x\Diamond_a\top \wedge \Diamond_b(\Diamond_b\top \wedge \Box_b\Diamond_a\top)$) and s_2' is modally distinguishable from s_2 (in $\mathcal{M} * U_1$) then s_2' is propositionally distinguishable from s_2.

Furthermore, we can show that s_2 and s_2' also have to be propositionally distinguishable if they are modally distinguishable in $\mathcal{M}$ (as opposed to $\mathcal{M} * U_1$). Suppose that $\mathcal{M} * U_1$ matches the above diagram, and that s_2 is modally distinguishable from s_2' in $\mathcal{M}$. Let δ be a modal formula that distinguishes between s_2 and s_2' (in $\mathcal{M}$), and assume without loss of generality that δ holds on s_1. Note that we previously concluded that $\mathcal{M}, s_2 \models \Diamond_b\top$ and that, since s_2' is a b-successor of a state $s' \in [s_0]_{\mathcal{M}}$, also $\mathcal{M}, s_2' \models \Diamond_b\top$.

We now distinguish between 3 cases.

- Suppose $\mathcal{M} * U_1, s_2 \models \Box_b\bot$ and $\mathcal{M} * U_1, s_2' \models \Box_b\bot$. Then let $U_1' := U_1 \cup \{\chi \wedge \neg p, b, \top\}$.
- Suppose one of s_2 and s_2' satisfies $\Box_b\bot$ while the other satisfies $\Diamond_b\top$. Then let $U_1' := U_1$.
- Suppose $\mathcal{M} * U_1, s_2 \models \Diamond_b\top$ and $\mathcal{M} * U_1, s_2' \models \Diamond_b\top$. Then let U_1' be the update obtained by replacing every clause $(\varphi, b, \psi) \in U_1$ by $(\varphi \wedge (p \vee \delta), b, \psi)$.

In any of the three cases, $\mathcal{M} * U_1'$ matches the diagram, and s_2 is distinguishable from s_2' in $\mathcal{M} * U_1'$ by the formula $\Diamond_b\top$. So s_2 and s_2' are propositionally distinguishable.

Summarizing again: $propd_x$ guarantees that if $\mathcal{M} * U_1$ matches the above diagram (i.e. $\mathcal{M}*U_1, s_1 \models \Box_a\bot\wedge\Diamond_x\Diamond_a\top\wedge\Diamond_b(\Diamond_b\top\wedge\Box_b\Diamond_a\top)$) and s_2 is modally distinguishable from s_2' (in $\mathcal{M}$), then s_2 is propositionally indistinguishable from s_2'.

Now, consider the formula $return_x$. It states that

$$\mathcal{M}, s \models \Box_b\langle\updownarrow\rangle(\Box_a\bot \wedge \Diamond_x\Diamond_a\top \wedge \Diamond_b(\Diamond_b\top \wedge \Box_b\Diamond_a\top)\wedge$$
$$[\updownarrow](\Diamond_x\Diamond_a\top \rightarrow \Box_b\Box_b\Diamond_a\top))$$

Once again, the initial $\Box_b$ operator takes us to any b-successor s_1 of s. Then, there is some update U_1 such that

$$\mathcal{M} * U_1, s_1 \models \Box_a\bot \wedge \Diamond_x\Diamond_a\top \wedge \Diamond_b(\Diamond_b\top \wedge \Box_b\Diamond_a\top)\wedge$$
$$[\updownarrow](\Diamond_x\Diamond_a\top \rightarrow \Box_b\Box_b\Diamond_a\top)$$

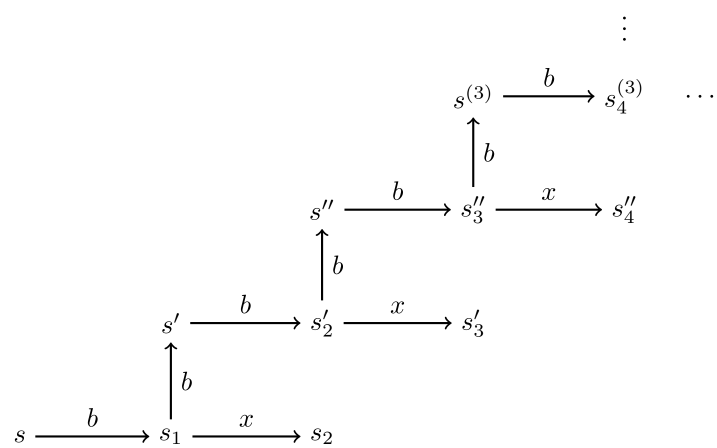
The first four conjuncts state that $\mathcal{M}*U_1$ matches the diagram drawn above. So if s_2 is modally distinguishable from some b-b-successor s_2' of s_1 in the model $\mathcal{M}$, then those two worlds are propositionally distinguishable. The final conjunct states that every arrow update U_2 that retains the x-arrow from s_1 to s_2 and the a-arrow from s_2 either removes one of the b-arrows between s_1 and s'_2 or retains the a-arrow on s'_2.

An update U_2 that removes the b-arrows between s_1 and s'_2 can be modified to an update $U'_2 := U_2 \cup \{(\top, b, \top)\}$ that retains the same a- and x-arrows as U_2, but retains all b-arrows. It follows that every U_2 that retains the x-arrow from s_1 to s_2 and the a-arrow from s_2 must also retain the a-arrow on s'_2. The state s_2 and s'_2 must therefore be modally indistinguishable (in $\mathcal{M} * U_1$). In particular, this implies that s_2 and s'_2 are propositionally indistinguishable. If s_2 and s'_2 had been modally distinguishable in $\mathcal{M}$ they would have been propositionally distinguishable, so it follows that s_2 and s'_2 are modally indistinguishable in $\mathcal{M}$.

In summary: *return*$_x$ guarantees that (in $\mathcal{M}$) there is a b-b-successor s'_2 of s_1 that is modally indistinguishable from s_2.

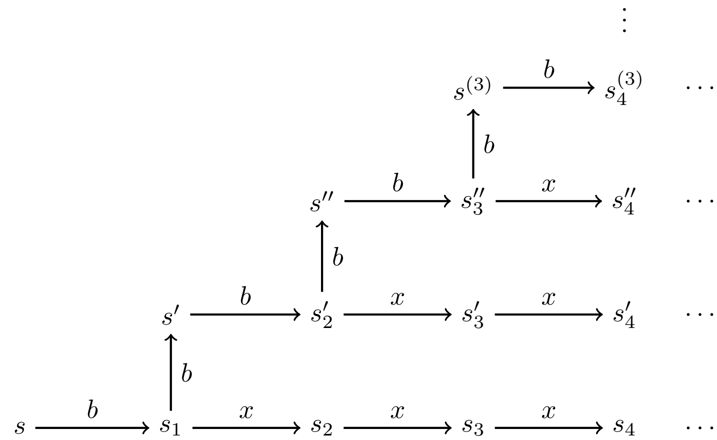
The state s' is a member of $[s_0]_{\mathcal{M}}$, so s'_2 a b-successor of a state that satisfies ψ_{grid}. All the conclusions that we drew about s_1 therefore also apply to s'_2: there is a unique, *up to modal indistinguishability* (*utmi*), x-successor s'_3 of s'_2, and this s'_3 is modally indistinguishable from a state s''_3 that is b-b-accessible from s'_2. The same procedure can be used on s''_3, and so on.

All in all, we get the following diagram,



where states with the same subscript are modally indistinguishable from one another.

Now, because s_2 is modally indistinguishable from s'_2, and s'_2 has a unique (utmi) x-successor s'_3, it follows that s_2 must have a unique (utmi) x-successor s_3 that is modally indistinguishable from s'_3, and therefore also from s''_3. Furthermore, s''_3 has a unique (utmi) x-successor s''_4, so s'_3 and s_3 have unique (utmi) x-successors s_4 and s'_4, respectively, that are modally indistinguishable from s''_4. The diagram can therefore be completed to the following:



The important thing to note is that $s_1, s_2, s_3, \cdots$ form an infinite sequence of states that each have a unique (utmi) x-successor and that each of them is modeally indistinguishable from a b-successor of a state that is a member of $[s_0]_{\mathcal{M}}$.

Furthermore, this holds for any direction x. So s_1 has of four such successor sequences, one for each direction. Every state in these successor sequences is modally indistinguishable from a b-successor of some state $s^{(n)} \in [s_0]_{\mathcal{M}}$, so every such state has four successor sequences of its own.

Now, consider *inverse* and *commute*. The formula *inverse* guarantees that, for opposite directions x and y, the x-y-successor of s_n is indistinguishable from s_n. The formula *commute*, guarantees that, for perpendicular directions x and y, the x-y-successor of s_n is indistinguishable from its y-x-successor. The successor sequences of s_1 therefore form a $\mathbb{Z} \times \mathbb{Z}$ grid.

Now, finally, consider ψ_{types}. This is a purely modal formula and it holds on s_0, so it holds on every $s \in [s_0]_{\mathcal{M}}$. The formula ψ_{types} guarantees that the conditions of a tiling are locally satisfied on the immediate b-successors of s. Let s_1 be any b-successor of $s \in [s_0]_{\mathcal{M}}$. Then *one_tile* guarantees that s_1 has exactly one tile type, *one_color* guarantees that that every side of s_1 has at most one color, *tile_colors* guarantees that every side of s_1 has the appropriate color for a tile of its type and *tile_match* guarantees that the tile edges match,; i.e. that the north side of the current tile is the same as the south side of its *up*-successor, and similarly for the other directions.

Since every state in the grid is modally indistinguishable from a b-successor of some $[s_0]_{\mathcal{M}}$ state, the conditions of a tiling are locally satisfied in every world of the grid, so they are globally satisfied. So if $\mathcal{M}, s_0 \models \chi_{types}$, then $\mathbb{Z} \times \mathbb{Z}$ can be tiled using *types*. □

Theorem 1. *The satisfiability problem of* AAUL *is co-RE hard.*

Proof. Given an instance *types* of the tiling problem, the formula χ_{types} is computable. Furthermore, Lemmas 1 and 5 show that χ_{types} is satisfiable if and only if *types* can tile the plane. The tiling problem is known to be co-RE complete [5], therefore the satisfiability problem of $AAUL$ is co-RE hard. □

4. Conclusion

We have shown that the satisfiability of $AAUL$ is uncomputable, like that of similar logics such as $APAL$ [9], GAL and CAL [2]. It is not currently known whether the satisfiability problems of $AAUL$, $APAL$ and GAL are co-RE. Typically, one would show that a satisfiability problem is co-RE by providing an axiomatization for the logic, thereby showing the validities of the logic to be RE. However, while there are known axiomatizations for $AAUL$, $APAL$ and GAL, these axiomatizations are infinitary and therefore cannot be used to enumerate the valid formulas of the logics in question.[3]

One interesting direction for future research is therefore to determine whether the satisfiability problems of $AAUL$, $APAL$ and GAL are co-RE complete, and whether these logics admit finitary axiomatizations.

In principle, the proof that we gave for the undecidability of $AAUL$ applies only to the satisfiability problem when considered over the class of all Kripke models. We consider this to be the most important version of the satisfiability problem for $AAUL$, since the class of all Kripke models is the "natural habitat" of $AAUL$, see [8] for details. Still, the satisfiability problem for $AAUL$ with respect to smaller classes of models could be formulated. We believe that, with only minor modifications, the proof presented in this paper would also show that satisfiability of $AAUL$ is undecidable with respect to other common classes of models such as KD45 and S5.

Acknowledgements

We acknowledge support from ERC project EPS 313360. Hans van Ditmarsch is also affiliated to IMSc, Chennai, as research associate. Furthermore, we would like to thank an anonymous reviewer for providing several precise and helpful comments.

[1] Thomas Ågotnes, Philippe Balbiani, Hans van Ditmarsch, and Pablo Seban. Group announcement logic. *Journal of Applied Logic*, 8(1):62 – 81, 2010.

[2] Thomas Ågotnes, Hans van Ditmarsch, and Tim French. The undecidability of quantified announcements. *Studia Logica*, 104(4):597–640, 2016.

[3] Philippe Balbiani, Alexandru Baltag, Hans van Ditmarsch, Andreas Herzig, Tomohiro Hoshi, and Tiago de Lima. 'Knowable' as 'known after an announcement'. *Review of Symbolic Logic*, 1(3):205–334, 2008.

[3] Finitary axiomatizations for $APAL$ and GAL were proposed, in [3] and [1] respectively, but these were later shown to be unsound.

[4] Alexandru Baltag, Lawrence Moss, and Sławomir Solecki. The logic of public announcements, common knowledge, and private suspicions. In I. Gilboa, editor, *Proceedings of the 7th conference on Theoretical aspects of rationality and knowledge*, pages 43–56. Morgan Kaufmann Publishers Inc., 1998.

[5] Robert Berger. *The Undecidability of the Domino Problem.* Number 66 in Memoirs of the American Mathematical Society. 1966.

[6] Laura Bozzelli, Hans van Ditmarsch, Tim French, James Hales, and Sophie Pinchinat. Refinement modal logic. *Information and Computation*, 239:303–339, 2014.

[7] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic.* Springer, Berlin, 2007.

[8] Hans van Ditmarsch, Wiebe van der Hoek, Barteld Kooi, and Louwe B. Kuijer. Arbitrary arrow update logic. *Artificial Intelligence*, 242:80–106, 2017.

[9] Tim French and Hans van Ditmarsch. Undecidability for arbitrary public announcement logic. In C. Areces and R. Goldblatt, editors, *Advances in Modal Logic*, volume 7, pages 23–42, 2008.

[10] James Hales. Arbitrary action model logic and action model synthesis. In *2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 253–262, 2013.

[11] Barteld Kooi and Brian Renne. Arrow update logic. *Review of Symbolic Logic*, 4(4):536–559, 2011.

[12] Jan Plaza. Logics of public communication. In M.L. Emrich, M.S. Phifer, M. Hadzikadic, and Z.W. Ras, editors, *Proceedings of the Fourth International Symposium on Methodologies for Intelligent Systems, Poster Session Program*, pages 201–216, 1989. Reprinted as [13].

[13] Jan Plaza. Logics of public communication. *Synthese*, 158:165–179, 2007.

[14] Hao Wang. Proving theorems by pattern recognition – II. *Bell System Technical Journal*, 40(1):1–41, 1961.