A Study on Multi-Dimensional Products of Graphs and Hybrid Logics☆

Mario R. F. Benevides[a,b], L. Menasché Schechter[a,*]

[a] Department of Computer Science, Federal University of Rio de Janeiro, Brazil
[b] Systems and Computer Engineering Program, Federal University of Rio de Janeiro, Brazil

Abstract

In this work, we address some issues related to products of graphs and products of modal logics. Our main contribution is the presentation of a necessary and sufficient condition for a countable and connected graph to be a product, using a property called intransitivity. We then proceed to describe this property in a logical language. First, we show that intransitivity is not modally definable and also that no necessary and sufficient condition for a graph to be a product can be modally definable. Then, we exhibit a formula in a hybrid language that describes intransitivity. With this, we get a logical characterization of products of graphs of arbitrary dimensions. We then use this characterization to obtain two other interesting results. First, we determine that it is possible to test in polynomial time, using a model-checking algorithm, whether a finite connected graph is a product. This test has cubic complexity in the size of the graph and quadratic complexity in its number of dimensions. Finally, we use this characterization of countable connected products to provide sound and complete axiomatic systems for a large class of products of modal logics. This class contains the logics defined by product frames obtained from Kripke frames that satisfy connectivity, transitivity and symmetry plus any additional property that can be defined by a pure hybrid formula. Most sound and complete axiomatic systems presented in the literature are for products of a pair of modal logics, while we are able, using hybrid logics, to provide sound and complete axiomatizations for many products of arbitrary dimensions.

Keywords: Products of Graphs, Intransitivity, Hybrid Languages, Model-Checking, Products of Modal Logics, Axiomatic Systems

☆A preliminary version of this work was published in the proceedings of LSFA 2009 [1].

*Corresponding author

Email addresses: mario@cos.ufrj.br (Mario R. F. Benevides), luisms@dcc.ufrj.br (L. Menasché Schechter)

1. Introduction

In this work, we address some issues related to products of graphs and products of modal logics. In particular, our main goal is to define a necessary and sufficient condition for a graph to be a non-trivial product of other graphs and describe this condition in a logical language. We then use this logical characterization of products of graphs to obtain two other interesting results dealing with the computational complexity to verify whether a finite graph is a product and with the construction of sound and complete axiomatic systems for products of modal logics. We are especially interested in products of modal logics with dimension greater than 2, for which there are very few results in the literature [2].

So, our fundamental task in this work is to find a necessary and sufficient condition for a graph to be isomorphic to a (cartesian) product[1] of non-trivial graphs and to verify whether this condition can be expressed in a modal language or in a hybrid language.

In [3] and [2], three properties that are satisfied in graphs that are products are presented: *left commutativity, right commutativity and the Church-Rosser property*. However, although these properties, together with the *reverse Church-Rosser property*, are necessary for a graph to be a product, they are not sufficient (as illustrated by an example in [3]). There are graphs that satisfy left and right commutativity and the Church-Rosser and reverse Church-Rosser properties, but cannot be decomposed as a product of other graphs.

In this work, we introduce a new property called *intransitivity* that, together with the previous ones, form a necessary and sufficient condition for a countable and (weakly) connected graph to be a product. The proof of the necessity of these properties is fairly simple and is done directly, without the need to assume that the graph is countable or connected. On the other hand, the proof of the sufficiency is done in two steps. First, we prove that if a countable and connected graph satisfies the five properties stated above, then its components must satisfy a particular isomorphism. Then, we show that if a countable and connected graph satisfies intransitivity and its components satisfy this particular isomorphism, then the graph is a product.

Once we are able to define this necessary and sufficient condition, with the help of intransitivity, it is now time to describe it in a logical language. As products of graphs appear naturally as an extension of the usual Kripke semantics of modal logics, the natural choice is to try to describe this condition in a modal language.

The limits to the expressive power of basic modal languages are fairly well known. There are a series of standard results that state that frames that are "similar" in a number of ways must agree on the validity of formulas [4]. Using these techniques, we show that the property of intransitivity is not definable in a

[1] In graph theoretical terminology, a product of graphs would be called a *multi-graph*, since it has many distinct sets of edges. In the context of modal logics and Kripke semantics, this notational difference is often lost and all these structures are called simply *labeled graphs*.

basic modal language. In fact, we also show that no condition that is necessary and sufficient for a graph to be a product can be definable in a basic modal language.

Hybrid logics are extensions of modal logics that allow explicit references to individual states of a model. Their goal is to extend the expressive power of ordinary modal logics. Besides proposition symbols, they have a second set of atomic formulas, called *nominals*, which have the property of being satisfied at exactly one state [5, 6]. Using a hybrid language, we are able to build a formula that describes intransitivity.

We then use this logical characterization to obtain two other interesting results. First, we determine the computational complexity of testing, for a finite connected graph, whether it is a product. For this test, we use a model-checking algorithm to verify the formulas that describe each of the five properties that characterize a product: left and right commutativity, the Church-Rosser and reverse Church-Rosser properties and intransitivity.

Finally, we use this characterization of countable connected products to provide sound and complete axiomatic systems for a large class of products of modal logics. This class contains the logics defined by product frames obtained from Kripke frames that satisfy connectivity, transitivity and symmetry plus any additional property that can be defined by a pure hybrid formula. The reasons behind such restrictions on the frames are explained in details in section 7.

Products of graphs come up naturally as a possible extension of ordinary Kripke semantics to multi-dimensional modal logics. [3] presents a good textbook discussion of multi-dimensional modal logics and provides many examples of products of modal logics, where the semantics is built using products of graphs. Most of the sound and complete axiomatic systems for products of modal logics presented in the literature are for products of a pair of modal logics, while we are able, using hybrid logics, to provide sound and complete axiomatizations for many products of arbitrary dimensions.

The paper is organized as follows. In section 2, we introduce the definition of a product of graphs and present four properties related to this definition: left and right commutativity and the Church-Rosser and reverse Church-Rosser properties. We also introduce a new property called intransitivity. In section 3, we present the concept of graph decomposition and use it to prove that the five properties presented in the previous section form a necessary and sufficient condition for a countable and connected graph to be a product. Section 4 shows that the property of intransitivity is not definable in a basic modal language and that no necessary and sufficient condition for a graph to be a product can be definable in a basic modal language. In section 5, we extend the modal language of the previous section to a hybrid language and show that intransitivity can be expressed by a hybrid formula. In section 6, we determine the computational complexity of testing, through a modal checking algorithm, whether a finite connected graph is a product. In section 7, we present the notion of a product of modal logics and, using a hybrid language, provide sound and complete axiomatic systems to a large class of products of modal logics. We summarize our results and present potential future works in section 8.

2. Product of Graphs

In this section, we define the product of graphs, following [3] and [2].

Definition 2.1. *Given $n \geq 2$ directed graphs $G_i = \langle V_i, E_i \rangle$, $1 \leq i \leq n$, we define their product G, notation $G = G_1 \times G_2 \times \ldots \times G_n$, as the graph $G = \langle V_1 \times V_2 \times \ldots \times V_n, \overline{E}_1, \overline{E}_2, \ldots, \overline{E}_n \rangle$, where for each i, $1 \leq i \leq n$, $\overline{E}_i$ is a binary relation on $V_1 \times V_2 \times \ldots \times V_n$ such that*

$$\langle u_1, \ldots, u_n \rangle \overline{E}_i \langle v_1, \ldots, v_n \rangle \text{ iff } u_i E_i v_i \text{ and } u_k = v_k, \text{ for } k \neq i.$$

An important particular case of the above definition concerns the product of two graphs. In this case, instead of the subscripts 1 and 2, it is common to use the subscripts h and v. They refer to the geometrical intuition of horizontal and vertical accessibility relations.

Definition 2.2. *Given two directed graphs $G_1 = \langle V_1, E_1 \rangle$ and $G_2 = \langle V_2, E_2 \rangle$, we define their product G, notation $G = G_1 \times G_2$, as the graph $G = \langle V_1 \times V_2, E_h, E_v \rangle$, where for all $x, u \in V_1$ and $y, v \in V_2$*

1. $\langle x, y \rangle E_h \langle u, v \rangle$ *iff* $x E_1 u$ *and* $y = v$ *and*
2. $\langle x, y \rangle E_v \langle u, v \rangle$ *iff* $y E_2 v$ *and* $x = u$.

An example of a product graph is shown in figure 1.

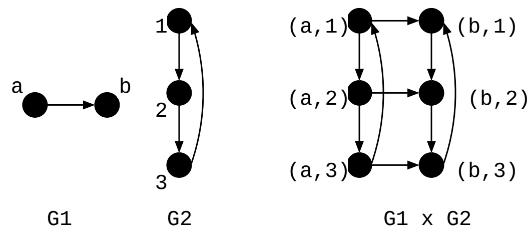

Figure 1: Product of Graphs

In this work, we would like to identify a necessary and sufficient condition for a graph to be a product of non-trivial graphs. A graph is said to be *trivial* if it has only one vertex and no edges. Every graph can be described as a product of itself with a trivial graph.

In [3] and [2], three properties that are satisfied in graphs that are products are presented. These properties, together with the reverse Church-Rosser property, are necessary for a graph to be a product (figure 2). Let $G = \langle V, \overline{E}_1, \ldots, \overline{E}_n \rangle$ and $1 \leq i, j \leq n$, $i \neq j$. These properties are defined as follows:

1. Left Commutativity: $\forall x, y, z \in V (x\overline{E}_j y \wedge y\overline{E}_i z \rightarrow \exists u \in V (x\overline{E}_i u \wedge u\overline{E}_j z))$
2. Right Commutativity: $\forall x, y, z \in V (x\overline{E}_i y \wedge y\overline{E}_j z \rightarrow \exists u \in V (x\overline{E}_j u \wedge u\overline{E}_i z))$
3. Church-Rosser Property: $\forall x, y, z \in V (x\overline{E}_j y \wedge x\overline{E}_i z \rightarrow \exists u \in V (y\overline{E}_i u \wedge z\overline{E}_j u))$

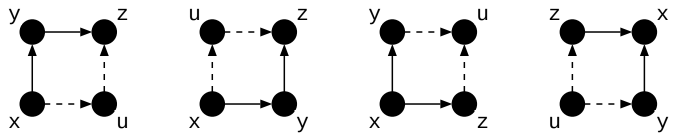

Figure 2: Left and Right Commutativity and Church-Rosser and Reverse Church-Rosser Properties

4. Reverse Church-Rosser Property: $\forall x, y, z \in V(y\overline{E}_j x \wedge z\overline{E}_i x \rightarrow \exists u \in V(u\overline{E}_i y \wedge u\overline{E}_j z))$

However, although these properties are necessary for a graph to be a product, they are not sufficient: there are graphs that satisfy left and right commutativity and the Church-Rosser and reverse Church-Rosser properties, but cannot be decomposed as a product of non-trivial graphs, as an example from [3] (figure 3) shows. The graph in figure 3 satisfies all of the four properties stated above, but it is not a product graph.

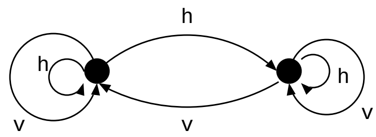

Figure 3: Counterexample to the sufficiency of the basic properties

It is not difficult to see that the graph in figure 3 is not a product. This graph has two sets of edges, so it could only be a product of two graphs. Besides that, as it has two vertices, it could only be a product between a graph with two vertices (let us call it G_1) and a graph with one vertex (G_2), as its set of vertices would be the cartesian product of the sets of vertices of G_1 and G_2. Now, the only non-trivial graph with only one vertex is a graph with one vertex and one loop edge, starting and ending in this vertex. It is not difficult to see, applying the definition of a product of two graphs, that the product between any graph (G_1) and this non-trivial graph with one vertex (G_2) is equal to G_1 with all its edges labeled with h plus one loop edge in every vertex, all of them labeled with v. So, if the graph in figure 3 were a product between such graphs G_1 and G_2, G_1 would be obtained by removing the v-loops from the graph in figure 3. However, such graph G_1 still contains two sets of edges, as there is a v-edge which is not a loop edge. Hence, the graph in figure 3 cannot be a product. Notice that our argument here was particular to the graph in figure 3 and cannot be easily generalized. This is a reason why it is important to identify a general necessary and sufficient condition for a graph to be a product of non-trivial graphs.

In order to obtain a necessary and sufficient condition we need to add a fifth property to the four stated before. We call it *intransitivity*.

Definition 2.3. *Let $G = \langle V, \overline{E}_1, \ldots, \overline{E}_n \rangle$ and $1 \leq i, j \leq n$, $i \neq j$. We say that G satisfies* intransitivity *if and only if every triple $\langle u, v, w \rangle$ of vertices of G that satisfies the conditions*

1. *$u \neq v$;*
2. *$v \neq w$;*
3. *there is an undirected path through edges of $\overline{E}_j$ from u to v and*
4. *there is an undirected path through edges of $\overline{E}_i$ from v to w*

also satisfies the following three conditions:

5. *$u \neq w$;*
6. *$\langle u, w \rangle \notin \overline{E}_i$;*
7. *$\langle u, w \rangle \notin \overline{E}_j$.*

Let xU_iy and xU_jy denote that there is an undirected path through edges of $\overline{E}_i$ ($\overline{E}_j$, respectively) from x to y. Intransitivity is illustrated in figure 4.

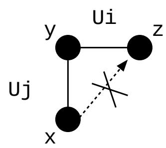

Figure 4: Intransitivity

Definition 2.3 lists the three conditions (items 5, 6 and 7) that we need for intransitivity. However, it turns out that they can be simplified, as, under the hypotheses in definition 2.3 (items 1 through 4), item 5 implies items 6 and 7. Suppose that all triples $\langle u, v, w \rangle$ that satisfy items 1 through 4 in definition 2.3 also satisfy item 5 ($u \neq w$). Now, suppose that there is one such triple $\langle a, b, c \rangle$ such that $\langle a, c \rangle \in \overline{E}_i$ (does not satisfy item 6). Then, aU_jb, bU_ic and $a\overline{E}_ic$. This implies that bU_ia. But then, $\langle a, b, a \rangle$ is a triple that satisfies items 1 through 4 but does not satisfy item 5 ($u \neq w$), which is a contradiction to our initial assumption. An analogous argument can be made regarding item 7 instead of item 6.

Thus, when we need to test whether a graph satisfies intransitivity, we just need to verify whether all triples that satisfy items 1 through 4 in definition 2.3 also satisfy item 5 in this same definition. On the other hand, when we know that a graph satisfies intransitivity, we may use any one of the conditions stated in items 5 through 7, according to our needs.

Following the above simplification, intransitivity can be described in the following way:

$$\forall x, y, z \in V((xU_jy \land yU_iz \land x \neq y \land y \neq z) \rightarrow (x \neq z))^2.$$

[2]It is important to notice that intransitivity cannot be expressed by a first order formula, since the definitions of U_i and U_j depend on transitive closures. Nevertheless, this property is still elementary, as it can be defined by a set of first order formulas.

3. Graph Decomposition

The problem of graph decomposition consists of, given a graph, to determine whether this graph can be decomposed in a product of non-trivial graphs. In this work, we consider a restricted version of this problem.

Problem 3.1. *Given a countable*[3]*, directed and weakly connected*[4] *(called just* connected *from now on) graph* $G = \langle V, \overline{E}_1, \ldots, \overline{E}_n \rangle$*, where* $\overline{E}_i \neq \emptyset$ *for all* $1 \leq i \leq n$*, determine whether* G *is isomorphic to a product* $G' = G_1 \times G_2 \times \ldots \times G_n$*, where* G_i *is non-trivial for all* $1 \leq i \leq n$*.*

In the general problem, the graph would not have to be necessarily countable or connected.

Hypothesis 3.2. *From now on, all the graphs* G *are considered to be directed, countable and connected and to be given in the form* $G = \langle V, \overline{E}_1, \ldots, \overline{E}_n \rangle$*.*

Remark 3.3. *We denote by* $\mathcal{V}(G)$ *the set of vertices of a graph* G*.*

In this section, we want to prove that a countable and connected graph G is a product if and only if it satisfies left and right commutativity, the Church-Rosser and reverse Church-Rosser properties and intransitivity. We start with the simpler direction.

Theorem 3.4. *If* G *is a product, then* G *satisfies left and right commutativity, the Church-Rosser and reverse Church-Rosser properties and intransitivity.*

Proof. We start with left commutativity. Let us take three vertices u, v and w of G such that $u\overline{E}_j v$ and $v\overline{E}_i w$. As G is a product $G_1 \times \ldots \times G_n$, $u = (u_1, \ldots, u_n)$, $v = (v_1, \ldots, v_n)$ and $w = (w_1, \ldots, w_n)$. Then, as $u\overline{E}_j v$, $u_k = v_k$, for all $k \neq j$, and $u_j E_j v_j$ and, as $v\overline{E}_i w$, $v_k = w_k$, for all $k \neq i$, and $v_i E_i w_i$. In particular, this means that, for $k \neq i$ and $k \neq j$, $u_k = v_k = w_k$. Now, take the vertex x such that $x_k = u_k$, for $k \neq i$ and $k \neq j$, $x_i = w_i$ and $x_j = u_j$ (this vertex exists, since $\mathcal{V}(G) = \mathcal{V}(G_1) \times \ldots \times \mathcal{V}(G_n)$). Then, as $u_k = v_k$, for $k \neq j$, $x_i = w_i$ and $v_i E_i w_i$, then $u_i E_i x_i$. This, together with $u_k = x_k$, for $k \neq i$, implies that $u\overline{E}_i x$. Now, as $u_j = x_j$, $v_k = w_k$, for $k \neq i$, and $u_j E_j v_j$, then $x_j E_j w_j$. This, together with $x_k = w_k$, for $k \neq j$, implies that $x\overline{E}_j w$. Right commutativity and the Church-Rosser and reverse Church-Rosser properties follow by analogous arguments.

Now, suppose that G does not satisfy intransitivity. Then, we have vertices x, y and z, such that $x \neq y$, $y \neq z$, there is an undirected $\overline{E}_j$-path from x to y and an undirected $\overline{E}_i$-path from y to z and $x = z$. As G is a product, $x = (x_1, \ldots, x_n)$, $y = (y_1, \ldots, y_n)$ and $z = (z_1, \ldots, z_n)$. Then, as there is an undirected $\overline{E}_j$-path from x to y, then $x_k = y_k$, for all $k \neq j$ (*). Also, as there

[3] A graph is countable if its set of vertices is countable.

[4] A graph G is weakly connected if, for any pair of vertices u and v of G, there is an undirected path from u to v in G.

is an undirected $\overline{E}_i$-path from y to z, $y_k = z_k$, for all $k \neq i$. As $x = z$, then $x_k = z_k$, for all k, which means that $x_k = y_k$, for all $k \neq i$ (**). (*) and (**) together imply that the three vertices are the same, contradicting the fact that $x \neq y$ and $y \neq z$. □

Notice that, in this direction of the proof, we make no use of hypothesis 3.2. This means that theorem 3.4 holds for any graph G. Now, we proceed to prove the other direction.

Proposition 3.5. *If a graph satisfies left and right commutativity and the Church-Rosser and reverse Church-Rosser properties, it also satisfies the following properties:*

1. *Extended Left Commutativity:* $\forall x, y, z \in V(xU_j^l y \wedge yU_i^k z \rightarrow \exists u \in V(xU_i^k u \wedge uU_j^l z))$;
2. *Extended Right Commutativity:* $\forall x, y, z \in V(xU_i^k y \wedge yU_j^l z \rightarrow \exists u \in V(xU_j^l u \wedge uU_i^k z))$.

where $uU_i^k v$ *and* $uU_j^l v$ *denote that there is an undirected path through edges of* $\overline{E}_i$ *(*$\overline{E}_j$*, respectively) of length* k *(*l*) from* u *to* v*.*

Proof. These properties follow by a straightforward induction on the length of the paths, using one of the four hypotheses for each of the four possible cases of edge incidences in the "corner" vertices: horizontal and vertical inward (reverse Church-Rosser), horizontal inward and vertical outward (right commutativity), horizontal outward and vertical inward (left commutativity) and horizontal and vertical outward (Church-Rosser). □

Definition 3.6. *Let* $G = \langle V, \overline{E}_1, \ldots, \overline{E}_n \rangle$ *and let* $G_k = \langle V, \overline{E}_k \rangle$, $1 \leq k \leq n$, *be subgraphs of* G*. The* k-components *are the maximal connected subgraphs of* G_k, $\{G_k^1, G_k^2, \ldots\}$*. Notice that the set of* k*-components is countable, since* G *is countable.*

Just as we presented the components of dimension one of the graph in the above definition, we also need to define the components of *co-dimension* one. Notice that in the particular case of bidimensional products, these definitions are equivalent.

Definition 3.7. *Let* $G = \langle V, \overline{E}_1, \ldots, \overline{E}_n \rangle$ *and let* $\widetilde{G}_k = \langle V, \overline{E}_1, \ldots, \overline{E}_{k-1}, \overline{E}_{k+1}, \ldots, \overline{E}_n \rangle$, $1 \leq k \leq n$, *be subgraphs of* G*. The* $\widetilde{k}$-components *are the maximal connected subgraphs of* $\widetilde{G}_k$, $\{\widetilde{G}_k^1, \widetilde{G}_k^2, \ldots\}$*. Notice that the set of* $\widetilde{k}$*-components is also countable, since* G *is countable.*

Definition 3.8. *We denote by* $\overline{E}_{\widetilde{k}}$ *the set of edges* $\overline{E}_1 \cup \ldots \cup \overline{E}_{k-1} \cup \overline{E}_{k+1} \cup \ldots \cup \overline{E}_n$*.*

A basic aspect about k-components and $\widetilde{k}$-components that needs to be noticed is that if you are on a vertex u in some k-component and you go through a path of edges in $\overline{E}_k$, you remain in the same k-component and if you are on a vertex u in some $\widetilde{k}$-component and you go through a path of edges in $\overline{E}_{\widetilde{k}}$, you remain in the same $\widetilde{k}$-component.

Proposition 3.9. *Two distinct k-components (or two distinct $\widetilde{k}$-components) have no vertices in common.*

Proof. We show the proof for $\widetilde{k}$-components. The proof for k-components is entirely analogous. Suppose that there are a vertex u and a pair of distinct $\widetilde{k}$-components $\widetilde{G}^i_k$ and $\widetilde{G}^j_k$ such that $u \in \widetilde{G}^i_k$ and $u \in \widetilde{G}^j_k$. Then, as the $\widetilde{k}$-components are maximal connected with respect to all the edges $\overline{E}_l$ such that $l \neq k$, every vertex $x \neq u \in \widetilde{G}^i_k$ has an undirected path from it to u through edges of $\overline{E}_{\widetilde{k}}$. Similarly, every vertex $y \neq u \in \widetilde{G}^j_k$ has an undirected path from it to u through edges of $\overline{E}_{\widetilde{k}}$. But this means that there is an undirected path from x to y through edges of $\overline{E}_{\widetilde{k}}$, which implies that x and y are in the same $\widetilde{k}$-component, contradicting the hypothesis that $\widetilde{G}^i_k$ and $\widetilde{G}^j_k$ are distinct. $\square$

Remark 3.10. *From now on, every time that we need to consider a pair of k-components G^i_k and G^j_k or a pair of $\widetilde{k}$-components $\widetilde{G}^i_k$ and $\widetilde{G}^j_k$, the two components in the pair* do not *need to be distinct, unless explicitly mentioned.*

Definition 3.11. *For $1 \leq k, l \leq n$, $k \neq l$, we say that a k-component G^i_k is a* l-neighbor to *the k-component G^j_k if there is an edge $\langle u, w\rangle \in \overline{E}_l$ such that $u \in G^i_k$ and $w \in G^j_k$. Notice that it is possible for a k-component to be a l-neighbor to itself.*

Definition 3.12. *For $1 \leq k, l \leq n$, $k \neq l$, let f^{ij}_{kl} be the (possibly partial and multi-valued) map that associates to each vertex $u \in G^i_k$ the set of vertices w such that $\langle u, w\rangle \in \overline{E}_l$ and $w \in G^j_k$. We say that f^{ij}_{kl} is the* l-induced map *from G^i_k to G^j_k.*

Proposition 3.13. *Let G be a graph that satisfies left and right commutativity, the Church-Rosser and reverse Church-Rosser properties and intransitivity and G^i_k be a l-neighbor to G^j_k. Then, the l-induced map f^{ij}_{kl} from G^i_k to G^j_k is an isomorphism.*

Proof. In this proof, $\mathrm{Dom}(f)$ and $\mathrm{Im}(f)$ denote, respectively, the *domain* and the *image* of a function f. Also, following remark 3.3, we denote by $\mathcal{V}(G)$ the set of vertices of a graph G.

1. f^{ij}_{kl} *is a function*: Suppose that there are vertices u, v and w, such that $v \neq w$, $u \in G^i_k$, $v, w \in G^j_k$ and $f^{ij}_{kl}(u) = \{v, w\}$ ($\langle u, v\rangle, \langle u, w\rangle \in \overline{E}_l$). If $u \neq v$, then we have an undirected $\overline{E}_l$-path from u to v, an undirected $\overline{E}_k$-path from v to w (since they are in the same k-component) and an edge from u to w, contradicting intransitivity. If $u = v$, then we have an undirected $\overline{E}_l$-path from v to w and an undirected $\overline{E}_k$-path from w to v, also contradicting intransitivity.
2. f^{ij}_{kl} *is injective*: Analogous to the previous item.

3. $\mathrm{Im}(f_{kl}^{ij}) = \mathcal{V}(G_k^j)$ (f_{kl}^{ij} is surjective): Let v in G_k^j. We need to find a vertex u in G_k^i such that $\langle u, v\rangle \in \overline{E}_l$. As G_k^i is a l-neighbor to G_k^j, there are vertices x in G_k^i and y in G_k^j such that $\langle x, y\rangle \in \overline{E}_l$. We may assume that $y \neq v$, otherwise the proof is over. Now, v and y are in G_k^j, so yU_kv. Then, by extended right commutativity, there is u such that xU_ku (which means that $u \in G_k^i$) and $u\overline{E}_lv$.
4. $\mathrm{Dom}(f_{kl}^{ij}) = \mathcal{V}(G_k^i)$ (f_{kl}^{ij} is total): Analogous to the previous item, using extended left commutativity instead.
5. $u\overline{E}_kw$ if and only if $f_{kl}^{ij}(u)\overline{E}_kf_{kl}^{ij}(w)$: First of all, $u\overline{E}_lf_{kl}^{ij}(u)$ and $w\overline{E}_l f_{kl}^{ij}(w)$. If $u\overline{E}_kw$, we can use left commutativity to conclude that $f_{kl}^{ij}(u)\overline{E}_k f_{kl}^{ij}(w)$. On the other hand, if $f_{kl}^{ij}(u)\overline{E}_kf_{kl}^{ij}(w)$, we can use right commutativity to conclude that $u\overline{E}_kw$.

□

Definition 3.14. *If G_k^i is a l-neighbor to G_k^j and the l-induced map f_{kl}^{ij} is an isomorphism between G_k^i and G_k^j, we call f_{kl}^{ij} a* l-primitive isomorphism.

Now, in a case such as the one in the above proposition, where all of the induced maps between neighbor k-components are isomorphisms, we can easily extend the isomorphisms beyond immediate neighbor k-components.

Let $L = [l_1, \ldots, l_n]$ denote a finite list. In order to define the composition of isomorphisms between k components, we extend the notation and write the maps as f_{kL}^{ij}. This notation can easily be used to denote the original l-induced maps and l-primitive isomorphisms, since in this case we just have to take $L = [l]$.

Remark 3.15. *If all the elements in the set $\{f_{kL_1}^{i,i+1}, f_{kL_2}^{i+1,i+2}, \ldots, f_{kL_{j-i}}^{j-1,j}\}$ are isomorphisms, then*

$$f_{kL}^{ij} = f_{kL_{j-i}}^{j-1,j} \circ \cdots \circ f_{kL_2}^{i+1,i+2} \circ f_{kL_1}^{i,i+1},$$

where $L = L_1 \circ L_2 \circ \cdots \circ L_{j-i}$, is also an isomorphism.

Remark 3.16. *If f_{kL}^{ij} is an isomorphism, then its inverse is also an isomorphism and is denoted by $f_{kL'}^{ji}$, where L' is the list symmetric to L.*

Definition 3.17. *If f_{kL}^{ij} is a primitive isomorphism or is obtained from primitive isomorphisms using composition and inverse, we call f_{kL}^{ij} an* orthogonal isomorphism *or* O-isomorphism.

Lemma 3.18. *Let G be a graph that satisfies left and right Commutativity, the Church-Rosser and reverse Church-Rosser properties and intransitivity. Then, for all pairs G_k^i and G_k^j of k-components, there is an O-isomorphism f_{kL}^{ij} between them. This means that all k-components are isomorphic between themselves and that the set of undirected paths through $\overline{E}_{\widetilde{k}}$-edges between G_k^i and G_k^j induce an isomorphism between the two k-components.*

Proof. First, as G is connected, for every k-component G_k^i there must be a k-component G_k^j such that either G_k^i is a neighbor to G_k^j or G_k^j is a neighbor to G_k^i. Then, as G is countable (in particular, as the number of k-components in G is countable), for any pair of k-components, we can build an O-isomorphism between them from the primitive isomorphisms given by proposition 3.13 using a *finite* number of inverse and composition operations on them, as described in remarks 3.16 and 3.15. □

Definition 3.19. *Let G be a graph. If G satisfies intransitivity and, for all pairs G_k^i and G_k^j of k-components, there is an O-isomorphism f_{kL}^{ij} between them, we say that G is* well-behaved.

Lemma 3.20. *Every $\widetilde{k}$-component contains as a subgraph at least one complete l-component, for all $l \neq k$.*

Proof. Let $\widetilde{G}_k^{i_k}$ be a $\widetilde{k}$-component and u be an arbitrary vertex in $\widetilde{G}_k^{i_k}$. It belongs to some l-component G_l^i. But all the vertices reachable from u through an undirected path of $\overline{E}_l$-edges, which are the vertices in the same l-component as u, are also in $\widetilde{G}_k^{i_k}$, if $l \neq k$. Hence, $\widetilde{G}_k^{i_k}$ contains as a subgraph at least one complete l-component. □

Lemma 3.21. *Let G be a well-behaved graph, G_k^i be a k-component and $\widetilde{G}_k^j$ be a $\widetilde{k}$-component. Then, $\mathcal{V}(G_k^i) \cap \mathcal{V}(\widetilde{G}_k^j)$ is a singleton set.*

Proof. First, we show that $\mathcal{V}(G_k^i) \cap \mathcal{V}(\widetilde{G}_k^j) \neq \emptyset$. Let u be a vertex in $\mathcal{V}(\widetilde{G}_k^j)$. It belongs to some k-component G_k^l. Using the O-isomorphism between G_k^i and G_k^l, there is a vertex w in $\mathcal{V}(G_k^i)$ such that there is an undirected path through $\overline{E}_{\widetilde{k}}$-edges from u to w in G. But this means that u and w belong to the same $\widetilde{k}$-component. Thus, w belongs to $\mathcal{V}(G_k^i) \cap \mathcal{V}(\widetilde{G}_k^j)$.

Now, if there were more than one vertex in the intersection $\mathcal{V}(G_k^i) \cap \mathcal{V}(\widetilde{G}_k^j)$, then there are at least two vertices x and y in G_k^i such that there are undirected paths through $\overline{E}_{\widetilde{k}}$-edges from u to x and from u to y in G (as $u, x, y \in \widetilde{G}_k^j$). But this contradicts the fact that undirected paths through $\overline{E}_{\widetilde{k}}$-edges induce an O-isomorphism between k-components. □

Corollary 3.22. *Let G be a well-behaved graph, H be a subgraph of G that contains as a subgraph at least one complete k-component and $\widetilde{G}_k^i$ be a $\widetilde{k}$-component. Then, $\mathcal{V}(H) \cap \mathcal{V}(\widetilde{G}_k^i) \neq \emptyset$.*

Proof. Let G_k^l be a k-component such that H contains G_k^l. From lemma 3.21, $\mathcal{V}(G_k^l) \cap \mathcal{V}(\widetilde{G}_k^i)$ is a singleton set. Then, as $\mathcal{V}(G_k^l) \subseteq \mathcal{V}(H)$, we have that $\mathcal{V}(H) \cap \mathcal{V}(\widetilde{G}_k^i) \neq \emptyset$. □

Corollary 3.23. *Let G be a well-behaved graph, $S = \{\widetilde{G}_k^{i_k} : 1 \leq k \leq n\}$ be a set that contains one $\widetilde{k}$-component for each $1 \leq k \leq n$. Then, $\bigcap\{\mathcal{V}(H) : H \in S\} \neq \emptyset$.*

Proof. Let $V_j = \bigcap\{\mathcal{V}(\widetilde{G}_k^{i_k}) : 1 \leq k \leq j+1\}$. We start with $V_1 = \mathcal{V}(\widetilde{G}_1^{i_1}) \cap \mathcal{V}(\widetilde{G}_2^{i_2})$. $\widetilde{G}_1^{i_1}$ is a maximal connected subgraph of G and, by lemma 3.20, contains as subgraph at least one complete l-component, for every $l \neq 1$. In particular, it contains at least one complete 2-component. Then, by corollary 3.22, $V_1 \neq \emptyset$.

Now, for $1 < k < n$, $V_k = V_{k-1} \cap \mathcal{V}(\widetilde{G}_{k+1}^{i_{k+1}})$. Let u be an arbitrary vertex in V_{k-1}. It belongs to some $(k+1)$-component G_{k+1}^i. But all the vertices reachable from u through an undirected path of $\overline{E}_{k+1}$-edges, which are the vertices in the same $(k+1)$-component as u, are also in the same $\widetilde{l}$-component that u is, if $l \neq k+1$. Then, the complete $(k+1)$-component that u belongs to is in every $\widetilde{G}_j^{i_j}$, for $1 \leq j \leq k$, which means that the subgraph of G generated by the vertices in V_{k-1} contains as a subgraph at least one complete $(k+1)$-component. Then, by corollary 3.22, $V_k \neq \emptyset$. □

Lemma 3.24. *Let G be a graph. If G satisfies intransitivity and, for all pairs G_k^i and G_k^j of k-components, there is an O-isomorphism f_{kL}^{ij} between them, then G is (isomorphic to) a product.*

Proof. For $1 \leq k \leq n$, let $G_k^* = \langle V_k, E_k \rangle$ be an arbitrary k-component with $V_k = \{v_k^1, k_k^2, \ldots\}$ (E_k is the restriction of the set $\overline{E}_k$ of G to V_k). Lemma 3.21 implies that each $\widetilde{k}$-component contains exactly one of the vertices in V_k. Besides that, by proposition 3.9, a vertex in V_k cannot be in more than one $\widetilde{k}$-component. Thus, without loss of generality, we can enumerate the vertices in V_k, such that $v_k^i \in \widetilde{G}_k^i$.

Let $P = \prod G_k^* = \langle \prod V_k, E_1^P, \ldots, E_n^P \rangle$. We want to prove that there is an isomorphism between G and P. Let us consider the map $\mathcal{L}$ that associates to each vertex $u \in \mathcal{V}(G)$ the vertex $(u_1, \ldots, u_n) \in \mathcal{V}(P)$ where $u_k = v_k^{i_k}$ if and only if $u \in \widetilde{G}_k^{i_k}$.

1. $\mathcal{L}$ is a well-defined total function: Clearly $\mathcal{L}$ is total, since every vertex u of G belongs to some $\widetilde{k}$-component $\widetilde{G}_k^{i_k}$, for each $1 \leq k \leq n$. Besides that, the map $\mathcal{L}$ associates to each vertex u of G, a unique vertex $(u_1, \ldots, u_n)$ of P. Otherwise, there would be, for at least one vertex u and at least one k, $1 \leq k \leq n$, a pair of distinct $\widetilde{k}$-components $\widetilde{G}_k^i$ and $\widetilde{G}_k^j$ such that $u \in \widetilde{G}_k^i$ and $u \in \widetilde{G}_k^j$. But this contradicts proposition 3.9.
2. If $u\overline{E}_k w$ in G, then $\mathcal{L}(u) E_k^P \mathcal{L}(w)$ in P: If $u\overline{E}_k w$, then, for all $l \neq k$, $u \in \widetilde{G}_l^{i_l}$ if and only if $w \in \widetilde{G}_l^{i_l}$. This means that if $\mathcal{L}(u) = (u_1, \ldots, u_n)$ and $\mathcal{L}(w) = (w_1, \ldots, w_n)$, then $u_l = w_l$, for all $l \neq k$. Now, we have that the vertex v_k^i of G belongs to $\widetilde{G}_k^i$, so if $u_k = v_k^j$ and $w_k = v_k^l$, this means that $u, v_k^j \in \widetilde{G}_k^j$ and $w, v_k^l \in \widetilde{G}_k^l$. Hence, there is an undirected path through $\overline{E}_{\widetilde{k}}$-edges in G from u to v_k^j and another undirected path through $\overline{E}_{\widetilde{k}}$-edges in G from w to v_k^l. Let G_k^i be the k-component that contains u and w. Using the O-isomorphism between G_k^i and G_k^*, as $u\overline{E}_k w$ in G_k^i, then $u_k = v_k^j \overline{E}_k v_k^l = w_k$ in G_k^*. This, together with $u_l = w_l$, for all $l \neq k$, implies that $\mathcal{L}(u) E_k^P \mathcal{L}(w)$.

3. If $\mathcal{L}(u)E_k^P w'$ in P, then there is a *unique* vertex w in G such that $w' = \mathcal{L}(w)$ and $u\overline{E}_k w$: Let $L(u) = (u_1, \ldots, u_n)$ and $w' = (w_1, \ldots, w_n)$. Then, as $\mathcal{L}(u)E_k^P w'$ in P, we have that $u_l = w_l$, for all $l \neq k$ and $u_k E_k w_k$. If $u_k = v_k^i$ and $w_k = v_k^j$, then, $v_k^i E_k v_k^j$ in G_k^*. Besides that, this means that u and v_k^i are in the same $\widetilde{k}$-component $\widetilde{G}_k^i$, which implies that there is an undirected path through $\overline{E}_{\widetilde{k}}$-edges in G from u to v_k^i. Let G_k^l be the k-component that contains u. The O-isomorphism between G_k^l and G_k^* relates u to v_k^i, so, as $v_k^i E_k v_k^j$ in G_k^*, then there is a vertex $x \in G_k^i$ such that $u\overline{E}_k x$ and the O-isomorphism between G_k^l and G_k^* relates x to v_k^j, which is equivalent to saying that there is an undirected path through $\overline{E}_{\widetilde{k}}$-edges in G from x to v_k^j. Now, the vertex x satisfies the properties of the vertex that we were looking for. First, $u\overline{E}_k x$. Second, this implies that $u \in \widetilde{G}_l^{i_l}$ if and only if $w \in \widetilde{G}_l^{i_l}$, for all $l \neq k$. Thus, if $\mathcal{L}(x) = (x_1, \ldots, x_n)$, then $u_l = x_l$, for all $l \neq k$. Third, as $x \in \widetilde{G}_k^j$, then $x_k = v_k^j$. Hence, $\mathcal{L}(x) = w'$. Now, suppose that there is another vertex $y \neq x$ that also satisfies these properties, i.e., $u\overline{E}_k y$ and $\mathcal{L}(y) = \mathcal{L}(x)$. But if $u\overline{E}_k y$, then u, x and y belong to to the same k-component G_k^l. However, by lemma 3.21, as there is only one vertex in the intersection of the set of vertices of a k-component and the set of vertices of a $\widetilde{k}$-component, x and y are in distinct $\widetilde{k}$-components, so $\mathcal{L}(y) \neq \mathcal{L}(x)$.
4. If $u'E_k^P\mathcal{L}(w)$ in P, then there is a *unique* vertex u in G such that $u' = \mathcal{L}(u)$ and $u\overline{E}_k w$: Analogous to the previous item.
5. $\mathcal{L}$ is surjective: Suppose that there is no u in G such that $L(u) = (x_1, \ldots, x_n)$. Let $x_k = v_k^{i_k}$. This is equivalent to saying that the intersection $\bigcap\{\mathcal{V}(\widetilde{G}_k^{i_k}) : 1 \leq k \leq n\}$ is empty, contradicting corollary 3.23.
6. $\mathcal{L}$ is injective: By item 5, as $\mathcal{L}$ is surjective, every vertex in P is of the form $\mathcal{L}(x)$ for some vertex x in G. For all $1 \leq k \leq n$, a vertex $\mathcal{L}(x)$ has at least one k-neighbor $\mathcal{L}(y)$ in P ($\mathcal{L}(x)E_k^P\mathcal{L}(y)$ or $\mathcal{L}(y)E_k^P\mathcal{L}(x)$). In order to see this, suppose that there is a vertex $\mathcal{L}(x)$ and a given k such that $\mathcal{L}(x)$ does not have any k-neighbors. Then, this vertex alone forms a k-component. However, as all k-components are isomorphic between themselves, then all k-components are isolated vertices, which means that there are no E_k^P-edges in P. Using item 2, this also means that there are no $\overline{E}_k$-edges in G. This contradicts the definition of problem 3.1. Now, suppose that there are two distinct vertices u and w in G such that $\mathcal{L}(u) = \mathcal{L}(w)$. Let $\mathcal{L}(x)$ be a k-neighbor of $\mathcal{L}(u)$ and $\mathcal{L}(y)$ be a l-neighbor of $\mathcal{L}(u)$. Then, using items 3 and 4, we have that x is a k-neighbor to both u and w and that y is a l-neighbor to both u and w. But this means that there are both undirected paths through $\overline{E}_k$-edges and through $\overline{E}_l$-edges from u to w, which contradicts intransitivity.

□

Theorem 3.25. *If G satisfies left and right commutativity, the Church-Rosser and reverse Church-Rosser properties and intransitivity, then G is a product.*

Proof. Straightforward from lemmas 3.18 and 3.24. $\square$

Theorem 3.26. *Let G be a graph. G is a product if and only if G satisfies left and right commutativity, the Church-Rosser and reverse Church-Rosser properties and intransitivity.*

Proof. Straightforward from theorems 3.4 and 3.25. $\square$

4. Modal Definability

In this section, we show that the property of intransitivity is not definable in a basic modal language. In fact, we also show that no condition that is necessary and sufficient for a graph to be a product can be definable in a basic modal language. Even though we restricted ourselves to countable and connected graphs in the previous section, this restriction is not necessary for the undefinability results presented in this section.

4.1. A Basic Modal Language

In this section, we define a modal language with a family of modal operators: $\Diamond_i$, $\Diamond_i^{-1}$ and $\blacklozenge_i$, for $1 \leq i \leq n$.

Definition 4.1. *Let us consider a modal language consisting of a set Φ of countably many proposition symbols, the boolean connectives $\neg$ and $\wedge$ and the modal operators $\Diamond_i$, $\Diamond_i^{-1}$ and $\blacklozenge_i$, for $1 \leq i \leq n$. The formulas are defined as follows:*

$$\varphi ::= p \mid \top \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond_i\varphi \mid \Diamond_i^{-1}\varphi \mid \blacklozenge_i\varphi,$$

where $p \in \Phi$.

We freely use the standard boolean abbreviations $\vee$, $\rightarrow$, $\leftrightarrow$ and $\perp$ and also the abbreviation $\widetilde{\Box}_i\varphi = \neg\widetilde{\Diamond}_i\neg\varphi$, where $\widetilde{\Diamond}_i \in \{\Diamond_i, \Diamond_i^{-1}, \blacklozenge_i\}$ and $\widetilde{\Box}_i$ is the correspondent $\Box$.

We now define the structures in which we evaluate our formulas: *frames* and *models*.

Definition 4.2. *A* frame *is a tuple $\mathcal{F} = (V, \{R_i\}_{1\leq i\leq n})$, where V is a set (finite or not) of vertices and R_i, $1 \leq i \leq n$, are binary relations over V, i.e., $R_i \subseteq V \times V$. We also define the auxiliary relations U_i, $1 \leq i \leq n$, as the transitive closures of the relations $R_i \cup R_i^{-1}$.*

As we can see, a frame is a graph with the distinct sets of edges R_i, $1 \leq i \leq n$. Also, a graph with an appropriate number of distinct sets of edges (one for each modality $\Diamond_i$ of the logic) can be used as a frame for the logic. So, in the rest of this work, the terms *graph* and *frame* are considered equivalent.

Definition 4.3. *A* model *is a pair $\mathcal{M} = (\mathcal{F}, \mathbf{V})$, where $\mathcal{F}$ is a frame and $\mathbf{V}$ is a valuation function mapping proposition symbols into subsets of V, i.e., $\mathbf{V} : \Phi \rightarrow \mathcal{P}(V)$.*

The notion of satisfaction is defined as follows:

Definition 4.4. *Let $\mathcal{M} = (\mathcal{F}, \mathbf{V})$ be a model. The notion of* satisfaction *of a formula φ in a model $\mathcal{M}$ at a vertex v, notation $\mathcal{M}, v \Vdash \varphi$, can be inductively defined as follows:*

1. $\mathcal{M}, v \Vdash p$ *iff* $v \in \mathbf{V}(p)$*;*
2. $\mathcal{M}, v \Vdash \top$ *always;*
3. $\mathcal{M}, v \Vdash \neg\varphi$ *iff* $\mathcal{M}, v \nVdash \varphi$*;*
4. $\mathcal{M}, v \Vdash \varphi_1 \wedge \varphi_2$ *iff* $\mathcal{M}, v \Vdash \varphi_1$ *and* $\mathcal{M}, v \Vdash \varphi_2$*;*
5. $\mathcal{M}, v \Vdash \Diamond_i\varphi$ *iff there is a* $w \in V$ *such that* vR_iw *and* $\mathcal{M}, w \Vdash \varphi$*;*
6. $\mathcal{M}, v \Vdash \Diamond_i^{-1}\varphi$ *iff there is a* $w \in V$ *such that* wR_iv *and* $\mathcal{M}, w \Vdash \varphi$*;*
7. $\mathcal{M}, v \Vdash \blacklozenge_i\varphi$ *iff there is a* $w \in V$ *such that* vU_iw *and* $\mathcal{M}, w \Vdash \varphi$*.*

If $\mathcal{M}, v \Vdash \varphi$ for every vertex v in a model $\mathcal{M}$, we say that φ is *globally satisfied* in $\mathcal{M}$, notation $\mathcal{M} \Vdash \varphi$. And if φ is globally satisfied in all models $\mathcal{M}$ of a frame $\mathcal{F}$, we say that φ is *valid* in $\mathcal{F}$, notation $\mathcal{F} \Vdash \varphi$. We also write $G \Vdash \phi$ to denote that the formula ϕ is valid in G when G is used as a frame for the logic. Finally, if φ is valid in every frame $\mathcal{F}$, we say that φ is valid, notation $\Vdash \varphi$.

When we say that a formula ϕ defines or describes some graph property, this means that a graph G has the desired property if and only if $G \Vdash \phi$.

As shown in [3] and [2], a graph satisfies left commutativity, right commutativity and the Church-Rosser property if the following formulas are valid in it for all pairs $1 \leq i, j \leq n$, $i \neq j$:

1. $com_{ij} = \Diamond_j\Diamond_i p \leftrightarrow \Diamond_i\Diamond_j p$ (left and right commutativity);
2. $chr_{ij} = \Diamond_i\Box_j p \rightarrow \Box_j\Diamond_i p$ (Church-Rosser property).

Then, a graph satisfies the reverse Church-Rosser property if the following formula, analogous to chr_{ij}, is valid in it for all pairs $1 \leq i, j \leq n$, $i \neq j$:

3. $rchr_{ij} = \Diamond_i^{-1}\Box_j^{-1}p \rightarrow \Box_j^{-1}\Diamond_i^{-1}p$.

4.2. A Limitative Result

The limits to the expressive power of basic modal languages are fairly well known. There are a series of standard results that state that frames that are "similar" in a number of ways must agree on the validity of formulas. We can then use these results to prove that a certain property *cannot* be expressed by any modal formula. To do this, we take two frames that are "similar" and show that in one the desired property holds, while in the other it does not. We present one of these "similarity" results (more details about it and other related results may be found in [4]), and then we prove two results for graph products using it.

Definition 4.5. *Let $\mathcal{M} = (W, \{R_i\}_{1\leq i\leq n}, \mathbf{V})$ and $\mathcal{M}' = (W', \{R'_i\}_{1\leq i\leq n}, \mathbf{V}')$ be two models. A function $f : W \rightarrow W'$ is a* bounded morphism *from $\mathcal{M}$ to $\mathcal{M}'$ if it satisfies the following conditions:*

1. w *and* $f(w)$ *satisfy the same proposition symbols;*

2. *f is a homomorphism with respect to R_i (if wR_iv, then $f(w)R'_i\, f(v)$);*
3. *if $f(w)R'_iv'$, then there is a v such that wR_iv and $f(v) = v'$;*
4. *if $w'R'_if(v)$, then there is a w such that wR_iv and $f(w) = w'$.*

A similar definition can be given for a bounded morphism of frames, just removing the part of the above definition that deals with valuations (item (i)). If there is a bounded morphism from a model (frame) $\mathcal{M}$ ($\mathcal{F}$) to a model (frame) $\mathcal{M}'$ ($\mathcal{F}'$), we use the notation $\mathcal{M} \to \mathcal{M}'$ ($\mathcal{F} \to \mathcal{F}'$). If there is a surjective bounded morphism, then we say that $\mathcal{M}'$ ($\mathcal{F}'$) is a bounded morphic image *of $\mathcal{M}$ ($\mathcal{F}$) and use the notation $\mathcal{M} \Rightarrow \mathcal{M}'$ ($\mathcal{F} \Rightarrow \mathcal{F}'$).*

The last item of the previous definition is usually not necessary. However, as the modalities $\Diamond_i^{-1}$ and $\blacklozenge_i$ deal with the inverses of the relations R_i, we have to enforce it to get the preservation result that we want. It may seem like conditions such as "if wU_iv, then $f(w)U'_i\, f(v)$", which is analogous to condition (ii), and others analogous to conditions (iii) and (iv) should also be added. However, this is not necessary, as the definition of U_x, with its use of transitive closure, and conditions (ii), (iii) and (iv) already imply such conditions.

Below is a basic theorem about modal definability that is going to be used to prove our results. Its proof for a language that contains only one modality can be found at [4]. It is not difficult to extend that proof to a language that contains a family of modalities, each with its accessibility relation.

Theorem 4.6. *Let $\mathcal{M}$ and $\mathcal{M}'$ be two models such that $\mathcal{M} \to \mathcal{M}'$. Then, $\mathcal{M}, w \Vdash \phi$ if and only if $\mathcal{M}', f(w) \Vdash \phi$.*

Corollary 4.7. *Let $\mathcal{F}$ and $\mathcal{F}'$ be two frames such that $\mathcal{F} \Rightarrow \mathcal{F}'$. If $\mathcal{F} \Vdash \phi$, then $\mathcal{F}' \Vdash \phi$.*

Theorem 4.8. *Neither intransitivity nor its negation are modally definable.*

Proof. In figure 5, let $f = \{(1, a), (2, b), (3, a), (4, b)\}$ and $g = \{(a, A), (b, A)\}$. It is straightforward to prove that f and g are surjective bounded morphisms. It is also not difficult to see that the first and third graphs respect intransitivity, while the second does not. By corollary 4.7, since neither intransitivity nor its negation are preserved under bounded morphic images, they are not modally definable. □

Theorem 4.9. *No necessary and sufficient condition for a graph* to be *a product or for a graph* not to be *a product can be modally definable.*

Proof. We take again the same bounded morphisms between the graphs in figure 5. It is not difficult to see that the first and third graphs are products while the second is not. By corollary 4.7, since neither the property of being a product nor the property of not being a product are preserved under bounded morphic images, they are not modally definable. □

This is not the only possible proof of theorem 4.9. However, as the counterexample used in theorem 4.8 could also be used in theorem 4.9 without any change, it was our choice to prove both theorems through the use of bounded morphic images.

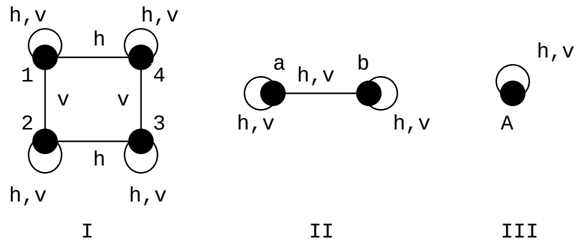

Figure 5: Graph III is a bounded morphic image of graph II, which is a bounded morphic image of graph I (each undirected edge represents a pair of symmetric edges)

5. A Hybrid Extension

As was shown in the previous section, a basic modal language does not have enough expressive power to describe the properties that we want. In order to achieve our goal, we need a language that is more expressive. In this section we describe a simple hybrid language and then use it to define intransitivity.

5.1. Language

A good way to improve the expressive power of a modal logic is to consider hybrid extensions of it. The fundamental resource that allows a logic to be called "hybrid" is a set of *nominals*. Nominals are a new kind of atomic symbol and they behave similarly to proposition symbols. The key difference between a nominal and a proposition symbol is related to their valuation in a model. While the set $\mathbf{V}(p)$ for a proposition symbol p can be any element of $\mathcal{P}(V)$, the set $\mathbf{V}(a)$ for a nominal a has to be a singleton set. This way, each nominal is true at exactly one state of the model, and thus, can be used to refer to this unique state. This is why these logics are called "hybrid": they are still modal logics, but they have the capacity to refer to specific states of the model, like in first-order logic.

The expressive power and computational complexity of a hybrid extension of a given modal logic usually lie between the ones of the original modal logic and the ones of first-order logic. This, however, depends on which operators, besides the nominals, are added to build the hybrid language. With the addition of state-variables and quantifiers, it is possible to achieve full first-order expressivity and complexity (undecidability). For a general introduction to hybrid logics, [5] and [6] can be consulted.

Here, we consider a simple hybrid extension of the modal logic presented in the previous section. We add nominals and the so-called satisfaction operators to the language.

Definition 5.1. *Let us consider a hybrid language consisting of a set Φ of countably many proposition symbols and a set Ω of countably many nominals such that $\Phi \cap \Omega = \emptyset$, the boolean connectives $\neg$ and $\wedge$, the modal operators $\Diamond_i$,*

$\Diamond_i^{-1}$ *and* $\blacklozenge_i$, $1 \leq i \leq n$ *and the satisfaction operators* $@_a$, *for each nominal* a. *The formulas are defined as follows:*

$$\varphi ::= p \mid a \mid \top \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Diamond_i\varphi \mid \Diamond_i^{-1}\varphi \mid \blacklozenge_i\varphi \mid @_a\varphi,$$

where $p \in \Phi$ *and* $a \in \Omega$.

It is common in the literature to use the letters i, j and k for nominals. However, as we already use these letters for the multiple edges of the graphs and frames, we choose to denote the nominals by the first letters a, b and c.

The definition of a frame for this language is the same as definition 4.2, but the definition of a model is slightly different from definition 4.3.

Definition 5.2. *A* hybrid model *is a pair* $\mathcal{M} = (\mathcal{F}, \mathbf{V})$, *where* $\mathcal{F}$ *is a frame and* $\mathbf{V}$ *is a valuation function mapping proposition symbols into subsets of* V, *i.e.,* $\mathbf{V} : \Phi \to \mathcal{P}(V)$ *and mapping nominals into singleton subsets of* V, *i.e, if* a *is a nominal then* $\mathbf{V}(a) = \{v\}$ *for some* $v \in V$. *We call this unique state that belongs to* $\mathbf{V}(a)$ *the* denotation *of* a *under* $\mathbf{V}$. *We can also say that* a *denotes the single state belonging to* $\mathbf{V}(a)$.

The notion of satisfaction is defined as follows:

Definition 5.3. *The notion of satisfaction is defined adding the following extra clauses to definition 4.4:*

1. $\mathcal{M}, v \Vdash a$ *iff* $v \in \mathbf{V}(a)$;
2. $\mathcal{M}, v \Vdash @_a\varphi$ *iff* $\mathcal{M}, d_a \Vdash \varphi$, *where* d_a *is the denotation of* a *under* $\mathbf{V}$.

5.2. Hybrid Definability

Using this hybrid language, we can now express intransitivity.

Theorem 5.4. *A graph* G *respects intransitivity if and only if* $G \Vdash int_{ij}$ *for all pairs* $1 \leq i, j \leq n$, $i \neq j$, *where* int_{ij} *is the formula*

$$int_{ij} = (a \wedge \neg b \wedge \blacklozenge_j(b \wedge \neg c \wedge \blacklozenge_i c)) \to \neg c.$$

Proof. ($\Leftarrow$) Suppose that $G \Vdash int_{ij}$ for all pairs $1 \leq i, j \leq n$, $i \neq j$, but G does not respect intransitivity. Then, there are at least three vertices x, y, z, $x \neq y$ and $y \neq z$, in G such that xU_jy, yU_iz and $x = z$. We evaluate int_{ij} in a model with a valuation $\mathbf{V}$ such that $\mathbf{V}(a) = \{x\}$, $\mathbf{V}(b) = \{y\}$ and $\mathbf{V}(c) = \{z\}$. Then, it is straightforward to see that $(G, \mathbf{V}), x \nVdash int_{ij}$, which contradicts the fact that int_{ij} is valid in G.

($\Rightarrow$) Suppose that G respects intransitivity but $G \nVdash int_{ij}$ for some pair $1 \leq i, j \leq n$, $i \neq j$. Then, there is a valuation $\mathbf{V}$ and a vertex u such that $(G, \mathbf{V}), u \nVdash int_{ij}$. Let $\mathbf{V}(a) = \{x\}$, $\mathbf{V}(b) = \{y\}$ and $\mathbf{V}(c) = \{z\}$. Then, we must have that $u = x$, $x \neq y$, $y \neq z$, xU_jy, yU_iz and $(G, \mathbf{V}), u \Vdash c$, which means that $u = x = z$. This contradicts the fact that G respects intransitivity. $\square$

It should be noticed that we do not need to use the satisfaction operators to describe intransitivity. Nevertheless, we keep the satisfaction operators in the language, as they tend to complement the nominals. While a nominal is a "name" for a specific state of the model, the satisfaction operators allow us to "jump" to different states of the model, using the "names" given by the nominals as references. This mutual relationship between nominals and satisfaction operators comes into play in the completeness proofs for hybrid logics. The satisfaction operators allow us to build completeness proofs for axiomatizations of some hybrid logics with a structure which closely resembles that of a Henkin-style completeness proof for first-order logic.[5] Then, as a direct consequence of this proof structure, we get the following result: if there is a model $\mathcal{M}$ and a vertex v in this model such that $\mathcal{M}, v \Vdash \phi$, where ϕ is a formula of such a hybrid logic, then there is a *countable model* $\mathcal{M}^*$ and a vertex v^* in this model such that $\mathcal{M}^*, v^* \Vdash \phi$. These results are explained in more details in section 7, where we deal with the axiomatization of hybrid logics. In particular, the result on countable models allows us to restrict our attention to this sort of model in our work in section 7. The reference [8] can also be consulted for more details on this subject.

6. Verification of the Product Property

In this section, we analyze the computational complexity to verify whether a finite connected graph is a product. This issue involves two basic decision problems.

Definition 6.1. *The* model-checking problem *consists of, given a formula ϕ and a finite model $\mathcal{M} = (W, R, \mathbf{V})$, determining the set $S_{\mathcal{M}}(\phi) = \{v \in W : \mathcal{M}, v \Vdash \phi\}$.*

Definition 6.2. *The* frame-checking problem *consists of, given a formula ϕ and a finite frame $\mathcal{F}$, determining whether $\mathcal{F} \Vdash \phi$.*

Definition 6.3. *We define the length of a formula φ, denoted by $|\varphi|$, inductively in the following way: $|p| = |a| = |\top| = 1$, $|\neg\phi| = |\Diamond_i\phi| = |\Diamond_i^{-1}\phi| = |\blacklozenge_i\phi| = |@_a\phi| = 1 + |\phi|$ and $|\phi_1 \wedge \phi_2| = 1 + |\phi_1| + |\phi_2|$.*

Definition 6.4. *Let $\mathcal{M} = (W, R, \mathbf{V})$ be a model. Let $|W|$ be the number of vertices in W and $|R|$ the number of pairs in R. We define the size of the model (or the frame, or the graph) as $|W| + |R|$.*

Proposition 6.5. *The model-checking problem for the logic presented in definition 5.1 is PTIME (linear) in the product of the size of the model and the length of the formula.*

[5] It is also technically possible to build such completeness proofs without the satisfaction operators, as shown in [7], but, with the presence of the satisfaction operators, the proofs become simpler, more direct and more elegant.

Proof. This is a consequence of the results about the model-checking of hybrid logics presented in [9]. In particular, we highlight, in that reference, the results presented in the first entry of table 1, the second item of theorem 4.3 and the comments regarding transitive modalities that follow the proof of theorem 4.3. □

We can provide a simple upper bound for the complexity of the frame-checking problem based on the complexity of the correspondent model-checking problem. We have that $\mathcal{F} \Vdash \phi$ if and only if $S_{\mathcal{M}}(\neg\phi) = \emptyset$ for every model $\mathcal{M}$ of $\mathcal{F}$. So, an algorithmic way to check whether $\mathcal{F} \Vdash \phi$ is to apply the model-checking algorithm to all the pairs $(\neg\phi, \mathcal{M})$, where $\mathcal{M}$ is a model of $\mathcal{F}$.

Thus, let FC be the complexity of the frame-checking problem and MC be the complexity of the model-checking problem. Then,

$$FC = O(2^{|p| \times m} \times m^{|a|} \times MC), \tag{1}$$

where $|p|$ is the number of distinct proposition symbols that occur in the given formula ϕ, $|a|$ is the number of distinct nominals that occur in ϕ and m is the number of vertices in $\mathcal{F}$. The distinction between proposition symbols and nominals in the above equation comes from the special restriction on the valuation of nominals. We need to apply the model-checking algorithm to every model $\mathcal{M}$ of the given frame $\mathcal{F}$. Every proposition symbol p that appears in ϕ may receive 2^m possible valuations $\mathbf{V}(p)$, while every nominal a may only receive m possible valuations $\mathbf{V}(a)$.

Proposition 6.6. *The frame-checking problem for the logic presented in definition 5.1 is PTIME (linear) in the length of the formula and EXPTIME in the size of the frame, in the number of distinct proposition symbols that occur in the formula and in the number of distinct nominals that occur in the formula.*

Proof. This result follows directly from the discussion above. □

It should be noticed that this calculation of the complexity of the frame-checking problem is just a general upper-bound and it can be reduced in some concrete situations.

From equation (1), we can see that, from the point of view of computational complexity, it is interesting to try to express the properties with formulas that use only nominals and no proposition symbols, since the presence of proposition symbols makes the verification of the property be exponential on the size of the frame.

Definition 6.7. *A* pure formula *is a formula with no occurrences of proposition symbols.*

So, pure formulas are interesting from the point of view of the complexity of the frame-checking problem. Besides that, as is shown in section 7, pure formulas also have advantages when used as axioms in an axiomatic system.

The formula in theorem 5.4, that describes intransitivity, is already pure. Now, we need to find pure formulas that describe left and right commutativity and the Church-Rosser and reverse Church-Rosser properties.

Proposition 6.8. *A graph G respects left and right commutativity if and only if $G \Vdash com^*_{ij}$ for all pairs $1 \leq i, j \leq n$, $i \neq j$, where com^*_{ij} is the formula*

$$com^*_{ij} = \Diamond_j \Diamond_i a \leftrightarrow \Diamond_i \Diamond_j a.$$

Proof. ($\Leftarrow$) Suppose that $G \Vdash com^*_{ij}$ for all pairs $1 \leq i, j \leq n$, $i \neq j$, but G does not respect at least one of left and right commutativity. We suppose that it does not respect left commutativity, as the symmetric case is entirely analogous. Then, there are at least a pair i and j and at least three vertices x, y, z in G such that xR_jy, yR_iz but there is no u such that xR_iu and uR_jz. We evaluate com^*_{ij} in a model with a valuation $\mathbf{V}$ such that $\mathbf{V}(a) = \{z\}$. Then, it is straightforward to see that $(G, \mathbf{V}), x \Vdash \Diamond_j \Diamond_i a$ but $(G, \mathbf{V}), x \nVdash \Diamond_i \Diamond_j a$, which contradicts the fact that com^*_{ij} is valid in G.

($\Rightarrow$) Suppose that G respects left and right commutativity but $G \nVdash com^*_{ij}$ for some pair $1 \leq i, j \leq n$, $i \neq j$. Then, there is a valuation $\mathbf{V}$ and a vertex u such that $(G, \mathbf{V}), u \nVdash com^*_{ij}$. So, either the left side of com^*_{ij} is satisfied at u but the right side is not or the other way around. We suppose that it is the first case, as the symmetric case is entirely analogous. Let $\mathbf{V}(a) = \{x\}$. Then, there is a vertex y such that uR_jy and yR_ix, but there is no vertex z such that uR_iz and zR_jx. This contradicts the fact that G respects left commutativity. $\square$

We can see that for left and right commutativity, the task of finding a pure formula that describes these properties was fairly easy, as we just have to substitute de propositional symbol p in the original formula com_{ij} by the nominal a. For the Church-Rosser property, this task is more difficult. The simple substitution of p by a in the original formula chr_{ij} does not work. However, it is possible to describe the Church-Rosser property with a pure formula. In [7], a pure formula for the Church-Rosser property is presented: $chr^*_{ij} = \Diamond_j a \rightarrow \Box_i \Diamond_j \Diamond_i^{-1} a$. It is also shown in [7] that it is not possible to describe the Church-Rosser property with a pure formula without the use of a converse modality $\Diamond_i^{-1}$.

Proposition 6.9. *A graph G respects the Church-Rosser property if and only if $G \Vdash chr^*_{ij}$ for all pairs $1 \leq i, j \leq n$, $i \neq j$, where chr^*_{ij} is the formula*

$$chr^*_{ij} = \Diamond_j a \rightarrow \Box_i \Diamond_j \Diamond_i^{-1} a.$$

Proof. ($\Leftarrow$) Suppose that $G \Vdash chr^*_{ij}$ for all pairs $1 \leq i, j \leq n$, $i \neq j$, but G does not respect the Church-Rosser property. Then, there are at least a pair i and j and at least three vertices x, y, z in G such that xR_iy, xR_jz but there is no u such that yR_ju and zR_iu. We evaluate com^*_{ij} in a model with a valuation $\mathbf{V}$ such that $\mathbf{V}(a) = \{z\}$. Then, it is straightforward to see that $(G, \mathbf{V}), x \Vdash \Diamond_j a$. Besides that, as there is no u such that such that yR_ju and zR_iu, we have that $(G, \mathbf{V}), x \Vdash \Diamond_i \Box_j \Box_i^{-1} \neg a$, which is equivalent to $(G, \mathbf{V}), x \nVdash \Box_i \Diamond_j \Diamond_i^{-1} a$. This contradicts the fact that chr^*_{ij} is valid in G.

($\Rightarrow$) Suppose that G respects the Church-Rosser property but $G \nVdash chr^*_{ij}$ for some pair $1 \leq i, j \leq n$, $i \neq j$. Then, there is a valuation $\mathbf{V}$ and a vertex u such that $(G, \mathbf{V}), u \nVdash chr^*_{ij}$. So, the left side of chr^*_{ij} is satisfied at u but the right side is not. Let $\mathbf{V}(a) = \{x\}$. Then, uR_jx. Besides that, there is a vertex y such that uR_iy and for all vertices z such that yR_jz it is not the case that xR_iz. This contradicts the fact that G respects the Church-Rosser property. $\square$

Proposition 6.10. *A graph G respects the reverse Church-Rosser property if and only if $G \Vdash rchr^*_{ij}$ for all pairs $1 \leq i, j \leq n$, $i \neq j$, where $rchr^*_{ij}$ is the formula*

$$rchr^*_{ij} = \Diamond_j^{-1} a \to \Box_i^{-1} \Diamond_j^{-1} \Diamond_i a.$$

Proof. The proof is analogous to the one from the previous proposition. $\square$

We now can determine how complex it is to test whether a finite connected graph is a product. By theorems 3.26 and 5.4 and propositions 6.8, 6.9 and 6.10, a finite connected graph G is a product if and only if $G \Vdash pro$, where pro is the formula

$$pro = \bigwedge_{1 \leq i,j \leq n, i \neq j} (com^*_{ij} \wedge chr^*_{ij} \wedge rchr^*_{ij} \wedge int_{ij})$$

As the test consists of, given a graph G, frame-check whether $G \Vdash pro$, we can calculate the complexity of this test in the following way. Let $FC(\phi)$ be the complexity to frame-check whether $G \Vdash \phi$. Then, we want to calculate $FC(pro)$. But, as there are $O(n^2)$ pairs $1 \leq i, j \leq n$, $i \neq j$ we have that

$$FC(pro) = O(n^2) \times FC_{ij},$$

where

$$FC_{ij} = FC(com^*_{ij}) + FC(chr^*_{ij}) + FC(rchr^*_{ij}) + FC(int_{ij}).$$

We should notice first that there are no proposition symbols in com^*_{ij}, chr^*_{ij}, $rchr^*_{ij}$ and int_{ij} and that the length of these formulas is constant and does not depend on the size of the graph. Besides that, com^*_{ij}, chr^*_{ij} and $rchr^*_{ij}$ have only one nominal and int_{ij} has three distinct nominals. Finally, as $\mathcal{F} \Vdash \phi$ if and only if $S_{\mathcal{M}}(\neg\phi) = \emptyset$ for every model $\mathcal{M}$ of $\mathcal{F}$, we check whether $\mathcal{F} \Vdash \phi$ by applying the model-checking algorithm to all the pairs $(\neg\phi, \mathcal{M})$, where $\mathcal{M}$ is a model of $\mathcal{F}$. Working with the negation of the formula $(\neg\phi)$, we can decrease the number of nominals involved in our present test, using the fact that the valuations of nominals are always singleton sets.

$$\neg int_{ij} \equiv \neg(\neg(a \wedge \neg b \wedge \blacklozenge_j(b \wedge \neg c \wedge \blacklozenge_i c)) \vee \neg c) \equiv a \wedge \neg b \wedge \blacklozenge_j(b \wedge \neg c \wedge \blacklozenge_i c) \wedge c$$

Now, a hybrid subformula of the form $a \wedge c$, where a and c are nominals, is satisfied in a vertex v of a model if and only if both nominals a and c are satisfied in v. Using the satisfaction definition for nominals, this means that $v \in \mathbf{V}(a)$ and $v \in \mathbf{V}(c)$. But the valuation of a nominal must always be a singleton set, which implies that $\mathbf{V}(a) = \mathbf{V}(c) = \{v\}$. Thus, if such a subformula must be satisfied in order for the whole formula to be satisfied, as in the case above, then we can rewrite the formula using only one of the two nominals:

$$\neg int_{ij} \equiv a \wedge \neg b \wedge \blacklozenge_j (b \wedge \neg c \wedge \blacklozenge_i c) \wedge c \equiv a \wedge \neg b \wedge \blacklozenge_j (b \wedge \neg a \wedge \blacklozenge_i a)$$

Using a similar argument, a hybrid subformula of the form $a \wedge \neg b$, where a and b are nominals, is satisfied in a vertex v of a model, if and only if a is satisfied in v, but not b. This means that $v \in \mathbf{V}(a)$ and $v \notin \mathbf{V}(b)$. As the valuations of nominals are singleton sets, this implies that there is no possible way for a and b to be satisfied at the same vertex in this model. Thus, if such a subformula must be satisfied in order for the whole formula to be satisfied, as in the case above, then any other subformula of the form $a \wedge \neg b$ or $\neg a \wedge b$ that appears in the same formula contains a redundant test and can be simplified to just a, in the first case, or b, in the second:

$$\neg int_{ij} \equiv a \wedge \neg b \wedge \blacklozenge_j (b \wedge \neg a \wedge \blacklozenge_i a) \equiv a \wedge \neg b \wedge \blacklozenge_j (b \wedge \blacklozenge_i a)$$

So, we can perform the test with only two nominals. Then, taking into account these observations and the formula in equation (1), we have that

$$FC_{ij} = O(m^2 \times MC),$$

where, in this case, MC is PTIME (in fact linear) in the size of the graph. This implies that

$$FC(pro) = O(n^2 \times m^2 \times MC).$$

Theorem 6.11. *The complexity to check whether a finite connected graph is a product using the above formula pro is PTIME (cubic) in the size of the graph and PTIME (quadratic) in the number of distinct sets of edges (number of dimensions).*

Proof. This result follows directly from the discussion above. □

7. Hybrid Axiomatizations for a Class of Products of Modal Logics

Products of graphs come up naturally as a possible extension of ordinary Kripke semantics to multi-dimensional modal logics. In this section, we present the concept of a *product of modal logics*, where the semantics is built using products of Kripke frames. First, we present the original definition, using modal languages. Then, we focus on the issues involved in the construction of complete axiomatic systems for such logics and present the limitations of modal languages to perform this task, especially in the case of products of dimension greater than two. We then use hybrid logic to bypass these limitations and build complete axiomatizations for a large class of products of modal logics.

7.1. Products of Modal Logics

Let Φ be a countable set of proposition symbols and Ω a countable set of nominals such that $\Phi \cap \Omega = \emptyset$. For a finite set of modal operators $\mathcal{O}$, we define the set of hybrid formulas $HFor(\mathcal{O})$ through the rule

$$\varphi ::= p \mid a \mid \top \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid O\varphi \mid @_a\varphi,$$

where $p \in \Phi$, $a \in \Omega$ and $O \in \mathcal{O}$. We also define the set of basic modal formulas $MFor(\mathcal{O})$ as the subset of $HFor(\mathcal{O})$ that contain only the formulas without nominals and satisfaction operators. When it is not relevant whether we are working with a hybrid language or with a basic modal language, we write $For(\mathcal{O})$ for the set of formulas over the language.

There are two common ways of defining a modal logic: the "syntactical" way and the "semantical" way. Let $F = For(\mathcal{O})$ be the set of all formulas in a given language (be it a hybrid language or a basic modal language). The syntactical way consists of taking a set $\mathcal{A} \subseteq F$ of axioms and a set $\mathcal{R}$ of rules and defining a modal logic L as the smallest set of formulas such that $\mathcal{A} \subseteq L$ and L is closed under the rules in $\mathcal{R}$. The semantical way consists of taking a class $\mathcal{C}$ of frames and defining a modal logic $L = \mathrm{Log}(F, \mathcal{C})$ as the set

$$\mathrm{Log}(F, \mathcal{C}) = \{\phi \in F : \mathcal{F} \Vdash \phi, \text{ for all } \mathcal{F} \in \mathcal{C}\}.$$

We can also define the duals of the sets $\mathrm{Log}(F, \mathcal{C})$. Let Fr_F be the class of frames in which formulas of F are evaluated and $\Sigma \subseteq F$ be a set of formulas. We define the class of frames $\mathrm{Fr}(F, \Sigma)$ as

$$\mathrm{Fr}(F, \Sigma) = \{\mathcal{F} \in Fr_F : \mathcal{F} \Vdash \phi, \text{ for all } \phi \in \Sigma\}.$$

If $L = \mathrm{Log}(F, \mathcal{C})$ is a semantically defined modal logic, we write $\Vdash_{\mathcal{C}} \phi$ to denote that $\phi \in L$.

Suppose that $\nVdash_{\mathcal{C}} \neg\phi$. This means that there is a frame $\mathcal{F} \in \mathcal{C}$ such that $\mathcal{F} \nVdash \neg\phi$. This, on the other hand, means that there is a model $\mathcal{M}$ of the frame $\mathcal{F}$ and a vertex v such that $\mathcal{M}, v \Vdash \phi$. So, ϕ is satisfied at a vertex of a model of a frame in $\mathcal{C}$.

Definition 7.1. *We say that a formula ϕ is* $\mathcal{C}$-satisfiable *if it is satisfied at a vertex of a model of a frame in $\mathcal{C}$ (equivalently, if $\nVdash_{\mathcal{C}} \neg\phi$). We say that a set Ψ of formulas is $\mathcal{C}$-satisfiable if there is a vertex of a model of a frame in $\mathcal{C}$ that satisfies all formulas $\phi \in \Psi$.*

Definition 7.2. *A modal logic L is called* Kripke-complete *if $L = Log(F, \mathcal{C})$ for some class $\mathcal{C}$ of frames. In this case, we say that L is characterized (or determined) by $\mathcal{C}$.*

Let $MFor_1 = MFor(\{\Diamond\})$ be the set of formulas over the basic mono-modal language. As a trivial example of a modal logic that is a subset of $MFor_1$ and can be defined in this semantical way, the set $\mathbf{K} = \{\phi \in MFor_1 : \Vdash \phi\}$ of all the valid formulas in $MFor_1$ is a modal logic, as $\mathbf{K} = \mathrm{Log}(MFor_1, \mathcal{C})$, where

$\mathcal{C}$ in this case is the class of *all* frames. The set $MFor_1$ itself is also a modal logic, as $MFor_1 = \text{Log}(MFor_1, \emptyset)$.

The notation **K** is usual in the modal logic literature for the logic over the basic mono-modal language determined by the class of all frames. Common notations for other modal logics that we use in this section are **K4**, **K5**, **T**, **B**, **D**, **Alt**, **B4** and **S5** for, respectively, the logics over the basic mono-modal language determined by the classes of transitive frames, euclidean frames, reflexive frames, symmetric frames, serial frames, functional frames, transitive and symmetric frames and transitive, symmetric and reflexive frames[6].

Let $MFor_n = MFor(\{\Diamond_1, \ldots, \Diamond_n\})$, $HFor_1 = HFor(\{\Diamond\})$ and $HFor_n = HFor(\{\Diamond_1, \ldots, \Diamond_n\})$ be the sets of formulas over the basic multi-modal language, the hybrid mono-modal language and the hybrid multi-modal language, respectively. If $L = \text{Log}(MFor_1, \mathcal{C})$, then we denote by $L(n), L(@)$ and $L(n, @)$ the sets $\text{Log}(MFor_n, \mathcal{C})$, $\text{Log}(HFor_1, \mathcal{C})$ and $\text{Log}(HFor_n, \mathcal{C})$, respectively. So, we have $\mathbf{K4}(@)$, $\mathbf{S5}(n)$ and so on.

A *product of modal logics* is a multi-modal logic that is defined semantically in the following way.

Definition 7.3. *Let $\{L_i : 1 \leq i \leq n\}$ be a finite set of Kripke-complete modal logics defined over the sets $For(\mathcal{O}_i)$, respectively, such that the sets of modalities $\mathcal{O}_i$, $1 \leq i \leq n$, are pairwise disjoint (in the present work, we consider that each L_i is mono-modal, having only one modality $\Diamond_i$). Let $F = For(\cup_i \mathcal{O}_i)$. We consider that $For(\mathcal{O}_i)$, $1 \leq i \leq n$, are either all sets of basic modal formulas or all sets of hybrid formulas. Then, the product of $L_1, \ldots, L_n$ is the multi-modal logic defined as*

$$L_1 \times \cdots \times L_n = Log(F, \mathcal{C}),$$

where

$$\mathcal{C} = \{\mathcal{F}_1 \times \cdots \times \mathcal{F}_n : \mathcal{F}_i \in Fr(L_i), \text{ for all } 1 \leq i \leq n\}.$$

Definition 7.4. *We denote by L^n the product $L \times \cdots \times L$, where L occurs n times.*

For example, $\mathbf{K} \times \mathbf{K}$ is the modal logic determined by the class of all product frames $\mathcal{F}_1 \times \mathcal{F}_2$, $\mathbf{K4} \times \mathbf{S5}$ is the modal logic determined by all product frames $\mathcal{F}_1 \times \mathcal{F}_2$ such that $\mathcal{F}_1$ is transitive and $\mathcal{F}_2$ is transitive, symmetric and reflexive and $\mathbf{S5}^n$ is the modal logic determined by all product frames $\mathcal{F}_1 \times \cdots \times \mathcal{F}_n$ such that each $\mathcal{F}_i$, $1 \leq i \leq n$ is transitive, symmetric and reflexive.

It should be noticed that the product of modal logics, being defined on Kripke-complete modal logics, is also Kripke-complete by definition.

7.2. The Axiomatization Problem for Products of Modal Logics

Now, one relevant issue about products of modal logics is the so-called *axiomatization problem.* Given the fact that a product $L = L_1 \times \cdots \times L_n$ is

[6]More details on these classes of frames can be found in [3] and [10].

semantically defined, is it also possible to find a correspondent syntactical definition for L?

In order to better define the axiomatization problem, we need the notions of soundness and completeness of an axiomatization. Let $\mathbb{A} = (\mathcal{A}, \mathcal{R})$ be an axiomatic system with the set of axioms $\mathcal{A}$ and the set of rules $\mathcal{R}$. Let $L_{\mathbb{A}}$ be the smallest set of formulas in the language under consideration such that $\mathcal{A} \subseteq L_{\mathbb{A}}$ and $L_{\mathbb{A}}$ is closed under the rules in $\mathcal{R}$. If $\phi \in L_{\mathbb{A}}$, we say that ϕ is a *theorem* of $\mathbb{A}$ and use the notation $\vdash_{\mathbb{A}} \phi$.

Definition 7.5. *We say that a formula ϕ is* $\mathbb{A}$-consistent *if $\nvdash_{\mathbb{A}} \neg\phi$. We say that a finite set Ψ_0 of formulas is* $\mathbb{A}$-consistent *if $\nvdash_{\mathbb{A}} \neg \bigwedge\{\phi : \phi \in \Psi_0\}$. Finally, we say that a set Ψ of formulas is* $\mathbb{A}$-consistent *if every finite subset $\Psi_0 \subseteq \Psi$ is $\mathbb{A}$-consistent.*

Definition 7.6. *If $\mathbb{A} = (\mathcal{A}, \mathcal{R})$ is an axiomatic system and Σ is a set of formulas, we denote by $\mathbb{A} + \Sigma$ the axiomatic system $(\mathcal{A} \cup \Sigma, \mathcal{R})$.*

Definition 7.7. *Let $L = Log(F, \mathcal{C})$ be a semantically defined modal logic and $\mathbb{A}$ an axiomatic system. We say that $\mathbb{A}$ is* sound *for L if and only if $\vdash_{\mathbb{A}} \phi$ implies $\Vdash_{\mathcal{C}} \phi$ for each formula ϕ in the language under consideration. Equivalently, as can be seen directly from the definitions of $\mathbb{A}$-consistency (definition 7.5) and $\mathcal{C}$-satisfiability (definition 7.1), $\mathbb{A}$ is* sound *for L if and only if every $\mathcal{C}$-satisfiable formula in the language under consideration is $\mathbb{A}$-consistent.*

Proposition 7.8. *$\mathbb{A}$ is* sound *for L if and only if every $\mathcal{C}$-satisfiable set of formulas in the language under consideration is $\mathbb{A}$-consistent.*

Proof. ($\Rightarrow$) Let Ψ be a $\mathcal{C}$-satisfiable set of formulas. Suppose that Ψ is not $\mathbb{A}$-consistent. Then, there are formulas $\phi_1, \dots, \phi_n \in \Psi$ such that $\vdash_{\mathbb{A}} \neg(\phi_1 \wedge \dots \wedge \phi_n)$. By soundness, $\Vdash_{\mathcal{C}} \neg(\phi_1 \wedge \dots \wedge \phi_n)$, which means that $\phi_1 \wedge \dots \wedge \phi_n$ is not $\mathcal{C}$-satisfiable. This implies that there is no vertex in no model of no frame in $\mathcal{C}$ that satisfies all of the formulas $\phi_1, \dots, \phi_n$. As all of these formulas are in Ψ, this contradicts the fact that Ψ is $\mathcal{C}$-satisfiable.

($\Leftarrow$) Let ϕ be a $\mathcal{C}$-satisfiable formula. Then, $\{\phi\}$ is a $\mathcal{C}$-satisfiable set. Now, by the hypothesis, $\{\phi\}$ is $\mathbb{A}$-consistent, which means that ϕ is $\mathbb{A}$-consistent, proving soundness. $\square$

Definition 7.9. *Let $L = Log(F, \mathcal{C})$ be a semantically defined modal logic and $\mathbb{A}$ an axiomatic system. We say that $\mathbb{A}$ is* complete *for L if and only if $\Vdash_{\mathcal{C}} \phi$ implies $\vdash_{\mathbb{A}} \phi$ for each formula ϕ in the language under consideration. Equivalently, $\mathbb{A}$ is* complete *for L if and only if every $\mathbb{A}$-consistent formula in the language under consideration is $\mathcal{C}$-satisfiable.*

In general, completeness does not automatically imply that every $\mathbb{A}$-consistent set of formulas in the language under consideration is $\mathcal{C}$-satisfiable. This is different from soundness, where we could jump from formulas to set of formulas (proposition 7.8).

Definition 7.10. *Let $L = Log(F, \mathcal{C})$ be a semantically defined modal logic and $\mathbb{A}$ an axiomatic system. We say that $\mathbb{A}$ is* strongly complete *for L if and only if every $\mathbb{A}$-consistent set of formulas in the language under consideration is $\mathcal{C}$-satisfiable.*

Strong completeness implies completeness (the argument is analogous to the one in the second part of the proof of proposition 7.8), but, in general, completeness does not imply strong completeness. As an example of a logic that has a complete axiomatic system, but does not have a strongly complete axiomatic system (unless we consider axiomatic systems with rules that can take an infinite number of premises), we can mention the dynamic modal logic PDL [4].

Then, for a product $L = L_1 \times \cdots \times L_n$, if we have a sound and complete axiomatic system $\mathbb{A}_i$ for each logic L_i, $1 \leq i \leq n$, is there a way to combine them in order to get a sound and complete axiomatic system for L?

7.3. Previous Results on the Subject

In the basic modal language, this issue is actually more complex than it may seem at first sight and there is no general method in the literature to take sound and complete axiomatizations of n arbitrary Kripke-complete logics and generate a sound and complete axiomatization for their product.

The problem seems to come from the fact that, as theorem 4.9 shows, no necessary and sufficient condition for a graph to be a product can be expressed in a basic modal language. If L_i is syntactically defined by the axiomatic system $\mathbb{A}_i = (\mathcal{A}_i, \mathcal{R}_i)$, for $1 \leq i \leq n$ and $C = \{com_{ij} \wedge chr_{ij} : 1 \leq i, j \leq n, i \neq j\}$, then $[L_1, \ldots, L_n]$ denotes the modal logic syntactically defined by $\mathbb{C} = (\cup_i \mathcal{A}_i, \cup_i \mathcal{R}_i) + C$. $[L_1, \ldots, L_n]$ is called the *commutator* of $L_1, \ldots, L_n$. As left and right commutativity and the Church-Rosser property are necessary conditions for a graph to be a product, we have that $[L_1, \ldots, L_n] \subseteq L_1 \times \cdots \times L_n$. However, as they are not sufficient, we do not have in general that $L_1 \times \cdots \times L_n \subseteq [L_1, \ldots, L_n]$. Logics for which this second inclusion is true are called *product-matching*.

The proof, for given logics L_i, $1 \leq i \leq n$ that $\phi \in L_1 \times \cdots \times L_n$ (which is a semantically defined logic) implies $\phi \in [L_1, \ldots, L_n]$ (which is a syntactically defined logic) is nothing more that a completeness proof for $\mathbb{C}$ with respect to $L_1 \times \cdots \times L_n$. The standard method for completeness proofs for modal logics is the construction of so-called *canonical models*. The vertices of the canonical model are all the possible maximal consistent sets of formulas of the language under consideration. The idea of the completeness proof using canonical models is that the canonical model should satisfy the semantical properties of the class $\mathcal{C}$ of frames that characterizes the logic. If this happens, then we can proceed to show that every $\mathbb{C}$-consistent formula is $\mathcal{C}$-satisfiable. For details on completeness proofs using canonical models, [4] should be consulted.

What happens in the specific case of commutators is that, as left and right commutativity and the Church-Rosser property are not sufficient for a graph to be a product, it is not possible to guarantee that the canonical model obtained from $\mathbb{C}$ is a product. So, canonical models are insufficient by themselves to derive completeness. Either another approach must be taken or, at least, some complementary steps must be done in order to reach completeness. Hence, there is no general method for completeness proofs of axiomatizations for products of logics, but there are particular logics that have been shown to be product-matching. As a start, $\mathbf{K} \times \mathbf{K}$ is product-matching, i.e., the axiomatic system of $[\mathbf{K}, \mathbf{K}]$ is complete for $\mathbf{K} \times \mathbf{K}$ [10].

An interesting result, shown in [10], states that for an arbitrary $n \in \mathbb{N}$, $n > 1$, the logic $\mathbf{Alt}^n$ is product-matching, i.e., the axiomatic system of the commutator of n copies of $\mathbf{Alt}$ is complete for $\mathbf{Alt}^n$.

Another result shown in [10] concerns the so-called PTC-logics. In order to define what a PTC-logic is, we first need the notions of a *closed* modal formula and of a *pseudo-transitive* modal formula. A *closed* modal formula is a modal formula without any proposition symbols and a *pseudo-transitive* modal formula is a modal formula of the form $\bigtriangledown \Box_k p \to \triangle p$, where $\bigtriangledown = \Diamond_{i_1} \ldots \Diamond_{i_r}$ is a sequence (possibly empty) of $\Diamond$ modalities, $\triangle = \Box_{j_1} \ldots \Box_{j_t}$ is a sequence (possibly empty) of $\Box$ modalities and p is a proposition symbol. For example, the following formulas are pseudo-transitive: $\Box p \to p$, $\Box p \to \Box\Box p$, $\Diamond\Box p \to p$ and $\Diamond_1 \Box_2 p \to p$. A PTC-formula is a formula which is either pseudo-transitive or closed. A PTC-logic is a modal logic axiomatized by an axiomatic system $\mathbb{P} = \mathbb{A}_{\mathbf{K}} + \Sigma$, where $\mathbb{A}_{\mathbf{K}}$ is a sound and complete axiomatic system for the logic $\mathbf{K}$ and Σ is a set of PTC-formulas. In [10], it is shown that, for any pair L_1 and L_2 of PTC-logics, the logic $L_1 \times L_2$ is product-matching, i.e., the axiomatic system of $[L_1, L_2]$ is complete for $L_1 \times L_2$. It is also shown in [10] that some non-PTC-logics fail to be product-matching. There are also logics for which it is unknown whether they are product-matching or not. Several of the most common logics are, in fact, PTC-logics, as $\mathbf{K4}$, $\mathbf{T}$, $\mathbf{B}$, $\mathbf{D}$ and $\mathbf{S5}$. For the detailed results, [10] should be consulted.

The completeness results for $\mathbf{K} \times \mathbf{K}$ and for a product of a pair of PTC-logics do not generalize to products of higher dimensions, as is shown by the following theorem from [11].

Theorem 7.11 ([11])**.** *Let $n \geq 3$ and let $L \subseteq MFor_n$ be any modal logic such that $\mathbf{K}^n \subseteq L \subseteq \mathbf{S5}^n$. Then, L is not finitely axiomatizable.*

7.4. Hybrid Axiomatizations

Summing up, while working in the basic modal language, we have no general method to build complete axiomatizations for products of logics and a significant group of products of logics of dimension greater than 2 cannot be axiomatized. As we already showed, a hybrid language offers the possibility to describe products of graphs, so it might also be well suited for an improvement in these results regarding axiomatizations for products of logics.

So, let us now consider logics $L_1 \times \cdots \times L_n$ over the hybrid language. First of all, $\mathbf{K}(n, @) \subseteq L_1 \times \cdots \times L_n$, so a good starting point to axiomatize $L_1 \times \cdots \times L_n$ is to get a complete axiomatization for $\mathbf{K}(n, @)$. In fact, this starting point will prove to be even better than expected, as completeness proofs for hybrid logics can be automatically extended to other hybrid logics under certain conditions, as shown in theorem 7.15.

We consider the set of axioms and rules shown in figure 6, where p and q are proposition symbols, a and b are nominals and φ and ψ are formulas.

(CT) All classical tautologies

(K$_i$) $\vdash \Box_i(p \to q) \to (\Box_i p \to \Box_i q)$

(Du$_i$) $\vdash \Box_i p \leftrightarrow \neg\Diamond_i \neg p$

(K$_@$) $\vdash @_a(p \to q) \to (@_a p \to @_a q)$

(SD) $\vdash @_a p \leftrightarrow \neg @_a \neg p$

(Ref) $\vdash @_a a$

(Agr) $\vdash @_a @_b p \leftrightarrow @_b p$

(Int) $\vdash a \to (p \leftrightarrow @_a p)$

(Back$_i$) $\vdash \Diamond_i @_a p \to @_a p$

(MP) If $\vdash \varphi$ and $\vdash \varphi \to \psi$, then $\vdash \psi$

(Gen$_i$) If $\vdash \varphi$, then $\vdash \Box_i \varphi$

(Gen$_@$) If $\vdash \varphi$, then $\vdash @_a \varphi$

(Nam) If $\vdash @_a \varphi$ and a does not occur in φ, then $\vdash \varphi$

(BG$_i$) If $\vdash @_a \Diamond_i b \to @_b \varphi$ and $b \neq a$ does not occur in φ, then $\vdash @_a \Box_i \varphi$

(Sub) If $\vdash \varphi$, then $\vdash \varphi^\sigma$, where σ is a substitution that uniformly replaces nominals by nominals and proposition symbols by arbitrary formulas

Figure 6: An axiomatic system $\mathbb{A}_{\mathbf{K}(n,@)}$ for the logic $\mathbf{K}(n,@)$

Definition 7.12. *We say that a hybrid model $\mathcal{M} = (\mathcal{F}, \mathbf{V})$ is a* named model *if, for every vertex v in the model, there is at least one nominal i in the hybrid language such that $\mathbf{V}(i) = \{v\}$.*

Proposition 7.13. *Every named model is countable.*

Proof. By hypothesis, the set of nominals in our hybrid languages is countable (definition 5.1). In a named model, each vertex of the model is the denotation of at least one nominal in the language. Thus, the cardinality of the set of vertices of a named model must be smaller than or equal to the cardinality of the set of nominals in the language, which means that there is a countable number of vertices in the model. $\square$

The combination of nominals with satisfaction operators that is present in hybrid languages allow us to build completeness proofs for axiomatizations of some hybrid logics that are very similar to a Henkin-style completeness proof for first-order logic. In particular, hybrid nominals play, in these completeness proofs, an analogous role to the one that is played by first-order constants in a completeness proof for first-order logic. An important observation about these Henkin-style completeness proofs is that they require the expansion of the original hybrid language with fresh nominals. However, as the original hybrid language is countable (it has countably many proposition symbols and countably many nominals, as stated in definition 5.1), only a *countable* set of fresh nominals needs to be added to the language during the proof. This proof structure allows us to prove completeness by building a very particular canonical model, which turns out to be a *named model*, as defined above. Then, following proposition 7.13, the models that are built in these completeness proofs are always countable. The reference [8] can also be consulted for more details on this subject.

Theorem 7.14. *The axiomatic system $\mathbb{A}_{\mathbf{K}(n,@)}$, shown in figure 6, is sound and strongly complete for the logic $\mathbf{K}(n,@)$. Moreover, each $\mathbb{A}_{\mathbf{K}(n,@)}$-consistent set of formulas is satisfiable in a* named *model.*

Proof. The proof of soundness and strong completeness for the mono-modal logic $\mathbf{K}(@)$ is well established in the literature. It is presented, in different levels of detail, in [4], [12] and [8]. The proof for the multi-modal case follows directly along the same lines. □

Theorem 7.15. *Let $\mathcal{C}$ be a class of frames defined as $\mathcal{C} = \{\mathcal{F} \in Fr_{HFor_n} : \mathcal{F} \Vdash \phi$ for all $\phi \in P\}$, where P is a set of pure formulas. Then, the axiomatic system $\mathbb{P} = \mathbb{A}_{\mathbf{K}(n,@)} + P$ is sound and strongly complete for the logic $Log(HFor_n, \mathcal{C})$. Moreover, each $\mathbb{P}$-consistent set of formulas is satisfiable in a* named *model.*

Proof. Again, as in the previous theorem, the proof for the mono-modal logic $\mathbf{K}(@)$ is well established in the literature. It is also presented in [4], [12] and [8] and the proof for the multi-modal case follows directly along the same lines. The key point in the proof is that the named model for a $\mathbb{P}$-consistent set of formulas is necessarily a model of a frame in $\mathcal{C}$. □

The theorem above provides automatic soundness and strong completeness for a large class of multi-modal hybrid logics. So, if we have a finite set P of pure formulas that defines the class of product frames, then all we need to do is add this set as axioms and we get completeness for $\mathbf{K}(@)^n$. Then, completeness for more restricted classes of product frames can also be obtained, as long as these classes can be described by pure formulas.

Theorem 3.26 describes a necessary and sufficient condition for a countable and connected graph to be a product and theorem 5.4 and propositions 6.8, 6.9 and 6.10 show that this condition can be expressed by a pure formula. Then, putting these results together with theorem 7.15, we can get some completeness results for products of modal logics under certain restrictions. These results are presented in theorem 7.26, which is built using theorem 7.15 and propositions 7.23 and 7.25. The reader may want to skip directly to this theorem and these propositions at a first reading, but it is important that he returns to this point of the text later on, since the following paragraphs explain why we also need some restrictions, as mentioned above.

We cannot just plug the pure formula defined by theorem 5.4 and propositions 6.8, 6.9 and 6.10 into theorem 7.15 and get the completeness that we want, as there are two issues that need to be addressed first.

We start by the most obvious one. There are two hypotheses that need to be satisfied before we can apply theorem 3.26: the graph needs to be connected and the graph needs to be countable. First, as the model built by theorem 7.15 is a named model, it is also countable (proposition 7.13). We get this result practically for free, just by using a hybrid logic with nominals and satisfaction operators.

Now, even though the first hypothesis, that the graph needs to be countable, is not a problem, the second hypothesis, that the graph needs to be connected, is a little more complicated. When working with a hybrid language, the named models built by the completeness proofs are not guaranteed to be connected. If we were working with ordinary modal logics, this issue could be solved by taking an appropriate generated submodel of the original model. This generated submodel would preserve the satisfaction of formulas and would also be connected (for more details on this preservation result related to generated submodels, [4] can be consulted).

However, in a hybrid logic, we need to refine our definition of "generated submodel" in order to ensure the validity of an analogous preservation result. In particular, as [7] and [13] show, in order to preserve the satisfaction of a set of formulas, a generated submodel of a hybrid model must contain all of the vertices of the original model that are the denotation of a nominal appearing in a formula in this set. This way, a generated submodel of our named model would be the standard generated submodel that we would build in the ordinary modal case with some extra vertices that are added following this condition on vertices that are the denotation of nominals. This generated submodel is also not guaranteed to be connected. Thus, we need to explicitly restrict ourselves to the class of connected frames and the best way to do this is to find a pure formula that describes connectivity.

Proposition 7.16. *A graph $G = G_1 \times \cdots \times G_n$ is connected if and only if G_i is connected for every $1 \leq i \leq n$.*

Proof. ($\Rightarrow$) Suppose that there is at least one G_i, $1 \leq i \leq n$, that is not connected. So, there is a pair of vertices $x, y \in G_i$ such that there is no undirected path between them in G_i. Now, take any two vertices $u = (u_1, \ldots, u_n)$ and $w = (w_1, \ldots, w_n)$ in G such that $u_k = w_k$, for all $k \neq i$, $u_i = x$ and $w_i = y$. Then, u and w belong to the same i-component and this i-component is not connected, because there would only be an undirected path through i-edges from u to w if there were an undirected path from x to y in G_i. As u_k can be chosen arbitrarily for $k \neq i$, all of the i-components of G are not connected. It is straightforward to see that this means that G itself is not connected.

($\Leftarrow$) Suppose that G_i is connected for every $1 \leq i \leq n$. Take two vertices $u = (u_1, \ldots, u_n)$ and $w = (w_1, \ldots, w_n)$ in G. We show how to find an undirected path between them in G. Let k_1 be the first coordinate such that $u_{k_1} \neq w_{k_1}$ (that is, $u_k = w_k$ for every $k \leq k_1$). Let $u_{k_1} = x$ and $w_{k_1} = y$. Now, $x, y \in G_{k_1}$ and, as G_{k_1} is connected, there is an undirected path in G_{k_1} between x and y. This implies that there is an undirected path through k_1-edges from u to $u' = (u'_1, \ldots, u'_n)$, where $u'_k = u_k$, for all $k \neq k_1$ and $u'_{k_1} = w_{k_1} = y$. Now, we repeat this process with the pair of vertices u' and w. Notice that if k_2 is the first coordinate such that $u'_{k_2} \neq w_{k_2}$, then $k_2 > k_1$. As n is finite, this process eventually stops in a vertex $u^* = w$. Then, if we attach together the undirected paths from u to u', from u' to u'' and so on, we get an undirected path from u to w. $\square$

Proposition 7.17. *A graph $G = (V, E)$ is connected if and only if $G \Vdash a \vee \blacklozenge a$.*

Proof. ($\Rightarrow$) Suppose that $G \nVdash a \vee \blacklozenge a$. Then, there is a valuation $\mathbf{V}$ and a vertex x such that $(G, \mathbf{V}), x \Vdash \neg a \wedge \blacksquare \neg a$. Let $\mathbf{V}(a) = y$. Then, we have that $x \neq y$ and that if there is an undirected path from x to a vertex z, then $z \neq y$. This means that G is not connected.

($\Leftarrow$) If $G \Vdash a \vee \blacklozenge a$, then $(G, \mathbf{V}), x \Vdash a \vee \blacklozenge a$ for every valuation $\mathbf{V}$ and every vertex x. Let $\mathbf{V}$ be such that $\mathbf{V}(a) = y \neq x$. Then, there is an undirected path between x and y in G. So, as the vertex x and the valuation $\mathbf{V}$ are arbitrary, this means that we have an undirected path between every pair of distinct vertices. Hence, G is connected. $\square$

Now, we address the second issue that prevents us from applying theorem 7.15. For this, we need to state the notion of *compactness*.

Definition 7.18 ([14])**.** *Let $L = Log(F, \mathcal{C})$ be a semantically defined modal logic. We say that L is* compact *if a set Σ of formulas is $\mathcal{C}$-satisfiable if and only if every finite subset $\Sigma_0 \subseteq \Sigma$ is $\mathcal{C}$-satisfiable.*

We need to add the modalities $\blacklozenge_i$ and $\Diamond_i^{-1}$ (as it is used in the definition of $\blacklozenge_i$) to the language, since the formulas that describe intransitivity and connectivity make use of them. However, when we add the modalities $\blacklozenge_i$ to the language, the modal logics lose compactness. For example, consider the set

$$\Sigma = \{\blacklozenge p, \neg\Diamond p, \neg\Diamond^{-1} p, \neg\Diamond\Diamond p, \neg\Diamond\Diamond^{-1} p, \neg\Diamond^{-1}\Diamond p, \neg\Diamond^{-1}\Diamond^{-1} p, \ldots\}.$$

Every finite set $\Sigma_0 \subseteq \Sigma$ is $\mathcal{C}$-satisfiable, where $\mathcal{C}$ in this case is the class of all frames. But Σ is not satisfiable in any vertex of any model of any frame.

The proof of completeness using canonical models gives strong completeness as its result (see theorems 7.14 and 7.15, for instance). The following proposition shows that if we have a sound and strongly complete axiomatic system for a logic $L = \text{Log}(F, \mathcal{C})$, then L is compact.

Proposition 7.19. *Let $L = Log(F, \mathcal{C})$ be a semantically defined modal logic and $\mathbb{A}$ an axiomatic system. If $\mathbb{A}$ is sound and strongly complete for L, then L is compact.*

Proof. First, we should notice that if a set Σ is $\mathcal{C}$-satisfiable, then trivially every finite subset $\Sigma_0 \subseteq \Sigma$ is also $\mathcal{C}$-satisfiable. So, in order to show compactness we only need to prove the other direction.

If every finite subset $\Sigma_0 \subseteq \Sigma$ is $\mathcal{C}$-satisfiable, then, by soundness, every finite subset $\Sigma_0 \subseteq \Sigma$ is $\mathbb{A}$-consistent. Then, by definition 7.5, this means that Σ is $\mathbb{A}$-consistent. Then, by strong completeness Σ is $\mathcal{C}$-satisfiable. $\square$

So, if a logic L is not compact, as the final result of completeness proofs using canonical models is strong completeness, such a proof will fail for L. This is a crucial observation about a limitation of proofs using canonical models that sometimes goes unnoticed. As [15] and [16] point out, this limitation comes from the so-called Lindenbaum's Lemma, one of the steps of the completeness proofs using canonical models. This lemma assumes compactness, even though this necessary hypothesis is often left implicit.

When the logic is not compact, sometimes it is still possible to build a completeness (but not strong completeness) proof using *finite canonical models* that are built from maximal consistent subsets of a finite set. However, for a completeness proof using finite canonical models to be successful, the logic must have the *finite model property*, i.e., every satisfiable formula must be satisfied in a vertex of a model of a *finite* frame. In our case, as we also need the canonical model to be a product, we would actually need the logic to have the so-called *product finite model property.*

Definition 7.20. *A product of modal logics has the* product finite model property *if every satisfiable formula is satisfied in a vertex of a model of a* finite *product frame.*

As the following theorem from [11] shows, many products of modal logics of dimension greater than 2 do not have the product finite model property.

Theorem 7.21 ([11])**.** *Let $n \geq 3$ and let $L \subseteq MFor_n$ be any modal logic such that $\mathbf{K}^n \subseteq L \subseteq \mathbf{S5}^n$. Then, L lacks the product finite model property.*

As $MFor_n \subset HFor_n$, the corollary below is immediate.

Corollary 7.22. *Let $n \geq 3$ and let $L \subseteq HFor_n$ be any modal logic such that $\mathbf{K}(@)^n \subseteq L \subseteq \mathbf{S5}(@)^n$. Then, L lacks the product finite model property.*

Theorem 7.21 and corollary 7.22 provide very strong negative results. They completely discourage any attempt to use finite canonical models to prove completeness for axiomatizations of products of modal or hybrid logics of arbitrary dimensions. Similarly, the lack of compactness, which is a consequence of the presence of the operators $\blacklozenge$, involving transitive closure of relations, makes the standard strong completeness proof for axiomatizations of products of logics impossible, at least in the general case.

In the simultaneous presence of these various strong negative results, we are left with only two options to try to axiomatize products of logics of arbitrary dimensions. The first option is to find a fragment of the class of products of logics where the use of transitive closures is unnecessary to describe the product property. This way, we may be able to describe this property without the modalities $\blacklozenge$, thus recovering compactness. The second option is to use infinitary proof systems (proof systems that contain infinitary rules, i.e., rules with a possibly infinite number of premises). The infinitary rules could allow us to bypass the lack of compactness and prove strong completeness. Such infinitary proof systems have already been developed for some hybrid logics, as in [17], [15] and [16].

From the start, our idea in the present work was to try to use the standard methods of hybrid completeness proofs to obtain a generic proof of completeness for axiomatizations of products of logics. Even though we have, so far, shown a series of negative results that prevent us from applying these standard methods to the class of all products of logics, we still think that we should try to finish what we started and see in which restricted cases of products of logics we can apply these standard methods of hybrid completeness proofs. Thus, in the rest of this section, we are going to work on the first option described above. However, the study of the second option described above is also very interesting as a possible future work, especially the study of infinitary sequent systems for hybrid logics, as described in [15] and [16].

So, in order to be able to continue, we need to find a way to express intransitivity and connectivity without the modalities $\blacklozenge$, so we can have compactness back and use theorem 7.15 to derive the completeness proofs. Let us restrict our attention to the class of transitive and symmetric frames. Over this class of frames, using symmetry, we have that $\Diamond_i^{-1}\phi \equiv \Diamond_i\phi$, which also implies (using transitivity) that $\blacklozenge_i\phi \equiv \Diamond_i\phi$. Besides that, the Church-Rosser property and the reverse Church-Rosser property are equivalent. So, over the class of transitive and symmetric frames, we can characterize the class of connected products with the pure formulas in figure 7.

(Con$_i$) $a \vee \Diamond_i a$

(Com$_{ij}$) $\Diamond_j\Diamond_i a \leftrightarrow \Diamond_i\Diamond_j a$

(Chr$_{ij}$) $\Diamond_j a \to \Box_i\Diamond_j\Diamond_i a$

(Int$_{ij}$) $(a \wedge \neg b \wedge \Diamond_j(b \wedge \neg c \wedge \Diamond_i c)) \to \neg c$

Figure 7: Pure formulas that characterize the class of connected products

Now, to finally get the result we are looking for, all that is left to do is to characterize the class of transitive and symmetric frames using pure formulas. In [4], the necessary formulas are presented. We show them in figure 8.

(4$_i$) $\Diamond_i\Diamond_i a \to \Diamond_i a$ (transitivity)

(B$_i$) $a \to \Box_i\Diamond_i a$ (symmetry)

Figure 8: Pure formulas that characterize the class of transitive and symmetric frames

Proposition 7.23. *Let P_{B4} be the set of pure formulas in figure 8 and let $\mathbb{A}_{\mathbf{B4}(n,@)}$ be the axiomatic system $\mathbb{A}_{\mathbf{B4}(n,@)} = \mathbb{A}_{\mathbf{K}(n,@)} + P_{B4}$. The axiomatic system $\mathbb{A}_{\mathbf{B4}(n,@)}$ is sound and strongly complete for the logic $\mathbf{B4}(n,@)$.*

Proof. This follows directly from theorem 7.15. $\square$

Definition 7.24. *Let $L = Log(F, \mathcal{C})$ be a semantically defined logic. Then, we denote by* $\mathbf{C}L$ *the logic defined by the class of connected frames in* $\mathcal{C}$.

Proposition 7.25. *Let P_{Prod} be the set of pure formulas in figure 7 and let $\mathbb{P}$ be the axiomatic system $\mathbb{P} = \mathbb{A}_{\mathbf{B4}(n,@)} + P_{Prod}$. The axiomatic system $\mathbb{P}$ is sound and strongly complete for the logic* $\mathbf{CB4}(@)^n$.

Proof. This follows directly from theorem 7.15. □

Theorem 7.26. *Let $L = L_1 \times \cdots \times L_n$ be a product of modal logics. If $\mathbf{CB4}(@) \subseteq L_i$, for all $1 \leq i \leq n$, and $\mathbb{A}_i = \mathbb{A}_{\mathbf{B4}(n,@)} + P_i$, where P_i is a finite set of pure formulas, is a sound and complete axiomatic system for L_i, then $\mathbb{A} = \mathbb{A}_{\mathbf{B4}(n,@)} + \sum_i P_i + P_{Prod}$ is a sound and strongly complete axiomatic system for L.*

Proof. This follows directly from theorem 7.15. □

As examples of properties that can be defined by pure formulas, we can mention reflexivity, irreflexivity, density, determinism, universality and trichotomy. Many of these properties cannot be defined by formulas in the basic modal language. So, theorem 7.26 provides sound and complete axiomatic systems for a large family of products of modal logics of arbitrary dimensions, while most of the results presented in the literature so far deal with bidimensional products.

As a last comment, what can we say about products of modal logics over the basic modal language, given the axiomatic systems that we built for products of modal logics over the hybrid language? First, as connectivity is not definable in the basic modal language [18], it is straightforward to see that $L = \mathbf{C}L$, if L is a modal logic over the basic modal language. Also, as the hybrid logics are *conservative extensions* of the corresponding modal logics obtained by excluding the nominals and satisfaction operators, we have that $L_1 \times \cdots \times L_n \subset \mathbf{C}L_1(@) \times \cdots \times \mathbf{C}L_n(@)$, which means that if we have a sound and complete axiomatic system $\mathbb{A}$ for $\mathbf{C}L_1(@) \times \cdots \times \mathbf{C}L_n(@)$, then all formulas $\phi \in L_1 \times \cdots \times L_n$ are also theorems of $\mathbb{A}$ ($\vdash_{\mathbb{A}} \phi$). As an example, we can get a sound and complete axiomatic system for $\mathbf{CS5}(@)^n$ using theorem 7.26. So, even though $\mathbf{S5}^n$ cannot be finitely axiomatized over the basic modal language, all of its formulas are deductible from the axiomatization of $\mathbf{CS5}(@)^n$. In fact, these formulas are exactly the subset of $\mathbf{CS5}(@)^n$ that contains only the formulas without nominals and satisfaction operators.

8. Conclusion

It is known that left and right commutativity and the Church-Rosser and reverse Church-Rosser properties are necessary conditions for a graph to be a non-trivial cartesian product of two other graphs, but their conjunction is not a sufficient condition. We introduce a new property called intransitivity, that, together with the former ones, form a necessary and sufficient condition for a countable and connected graph to be a product. We show that the property of intransitivity is not definable in a basic modal language. We also show that no condition that is necessary and sufficient for a graph to be a product can be definable in a basic modal language. We then extend our language to a hybrid language and show that in such a language we are able to express intransitivity.

We also determine the computational complexity of testing, for a finite connected graph, whether it is a product. For this test, we use a model-checking algorithm to verify the formulas that describe each of the five properties that characterize a product. We show that this test can be performed in polynomial time both in the size of the graph (cubic time) and in the number of dimensions (quadratic time).

Finally, we use the above characterization of countable connected products to provide sound and complete axiomatic systems for a large class of products of modal logics. While most of the sound and complete axiomatic systems for products of modal logics presented in the literature are for products of a pair of modal logics, we are able, using hybrid logics, to provide sound and complete axiomatizations for many products of arbitrary dimensions.

The work presented in [19] is another recent work that also deals with the product of Kripke frames in a hybrid logic setting. But its approach is different from ours. Instead of using the traditional hybrid language and the traditional semantical definition of a nominal as an atomic formula that must, by construction, denote an unique state of the model, it builds an extension of the hybrid language with a new semantical characterization of the nominals. The nominals as defined in [19] denote something like a sub-frame of co-dimension one inside the considered frame. The author restricts himself to two-dimensional frames, so each nominal would denote a “line” of the frame. With this definition, the language then uses two disjoint sets of nominals, one denoting the “horizontal lines” in the frame and the other denoting the “vertical lines”. As such, each state of the model is then denoted by an ordered pair of nominals, instead of a single nominal. If we consider n-dimensional frames, each state will then be denoted by an n-uple of nominals, taken from n disjoint sets. With this linguistic setup, the models in which the formulas are evaluated are already models based on product frames, so the technical difficulty presented in the study of products of Kripke frames, which is exactly how to filter the product frames from the class of all frames, is already bypassed.

The logic $\mathbf{S5}^n$ is related to the field of cylindric algebras [20, 21, 22]: the modal algebras corresponding to $\mathbf{S5}^n$ are the representable diagonal-free cylindric algebras of dimension n. As a future work, it would be interesting to analyze whether the axiomatization that we can provide for $\mathbf{S5}^n$ using the results in section 7 could be useful from the algebraic point of view.

In [23], products of logics are studied from the point of view of spatial and topological logics. In the last chapter of [23], the author develops a preliminary work on the possible use of hybrid logic for the study of spatial and topological logics and their products. As another future work, it would be interesting to analyze whether our results are useful or meaningful for the case of spatial and topological logics. The work presented in [24] could also be useful in this enterprise. This work deals with coalgebraic logics, in particular with the generalization of hybrid logics to coalgebraic hybrid logics. This generalization provides a series of completeness results for many hybrid logics. These completeness results could be useful, since a topological space is a special example of the notion of coalgebras.

Another interesting future work involves the study of infinitary proof systems for hybrid logics. The use of infinitary systems is a second way to bypass the lack of compactness that we face, in the general case, with our proposed axiomatizations. A line of work that seems particularly interesting is the study of infinitary sequent systems for hybrid logics, as described in [15] and [16], and their possible application to products of logics.

Finally, it would also be important to investigate alternative characterizations of products of connected graphs, especially if they do not require the use of transitive closures. That way, we might be able to use a compact logic, which would allow us to drop the restrictions of transitivity and symmetry in our hybrid axiomatizations.

Acknowledgments

The authors want to thank the anonymous reviewers for their valuable contributions. The authors received financial support from the research agencies CNPq, FAPERJ and CAPES.

References

[1] M. R. F. Benevides, L. M. Schechter, Product of graphs and hybrid logic, in: Proceedings of the 4th Workshop on Logical and Semantic Frameworks, with Applications (LSFA 2009), Vol. 256 of Electronic Notes in Theoretical Computer Science, Elsevier, 2009, pp. 103–118.

[2] A. Kurucz, Combining modal logics, in: P. Blackburn, J. van Benthem, F. Wolter (Eds.), Handbook of Modal Logics, Elsevier, 2006, pp. 869–924.

[3] D. Gabbay, A. Kurucz, F. Wolter, M. Zakharyaschev, Many-dimensional modal logics: theory and applications, Elsevier Science, 2003.

[4] P. Blackburn, M. de Rijke, Y. Venema, Modal Logic, Theoretical Tracts in Computer Science, Cambridge University Press, 2001.

[5] C. Areces, B. ten Cate, Hybrid logics, in: P. Blackburn, J. van Benthem, F. Wolter (Eds.), Handbook of Modal Logics, Elsevier, 2006, pp. 821–868.

[6] P. Blackburn, Representation, reasoning, and relational structures: a hybrid logic manifesto, Logic Journal of the IGPL 8 (3) (2000) 339–365.

[7] B. ten Cate, Model theory for extended modal languages, Ph.D. thesis, University of Amsterdam (2005).

[8] P. Blackburn, B. ten Cate, Pure extensions, proof rules, and hybrid axiomatics, Studia Logica 84 (2) (2006) 277–322.

[9] M. Franceschet, M. de Rijke, Model checking hybrid logics (with an application to semistructured data), Journal of Applied Logic 4 (3) (2006) 279–304.

[10] D. Gabbay, V. Shehtman, Products of modal logics, part 1, Logic Journal of the IGPL 6 (1) (1998) 73–146.

[11] R. Hirsch, I. Hodkinson, A. Kurucz, On modal logics between $\mathbf{K} \times \mathbf{K} \times \mathbf{K}$ and $\mathbf{S5} \times \mathbf{S5} \times \mathbf{S5}$, Journal of Symbolic Logic 67 (1) (2002) 221–234.

[12] P. Blackburn, M. Tzakova, Hybrid languages and temporal logic, Logic Journal of the IGPL 7 (1) (1999) 27–54.

[13] C. Areces, P. Blackburn, M. Marx, Hybrid logics: Characterization, interpolation and complexity, Journal of Symbolic Logic 66 (3) (2001) 977–1010.

[14] H. B. Enderton, A Mathematical Introduction to Logic, Academic Press, 2001.

[15] B. Kooi, G. Renardel de Lavalette, R. Verbrugge, Strong completeness for non-compact hybrid logics, in: Proceedings of 5th AiML Conference, 2004, pp. 212–223.

[16] B. Kooi, G. Renardel de Lavalette, R. Verbrugge, Hybrid logics with infinitary proof systems, Journal of Logic and Computation 16 (2) (2006) 161–175.

[17] S. Passy, T. Tinchev, An essay in combinatory dynamic logic, Information and Computation 93 (2) (1991) 263–332.

[18] M. R. F. Benevides, L. M. Schechter, Using modal logics to express and check global graph properties, Logic Journal of the IGPL 17 (5) (2009) 509–537.

[19] K. Sano, Axiomatizing hybrid products: how can we reason many-dimensionally in hybrid logic?, Journal of Applied Logic 8 (4) (2010) 459–474.

[20] Y. Venema, Many-dimensional modal logic, Ph.D. thesis, University of Amsterdam (1992).

[21] M. Marx, Y. Venema, Multi-dimensional Modal logic, Kluwer Academic Publishers, 1997.

[22] L. Henkin, J. D. Monk, A. Tarski, Cylindric Algebras, Parts I & II, North-Holland, 1971 & 1985.

[23] D. Sarenac, Products of topological modal logics, Ph.D. thesis, Stanford University (2006).

[24] R. Myers, D. Pattinson, L. Schröder, Coalgebraic hybrid logic, in: Proceedings of 12th FOSSACS Conference, Vol. 5504 of Lecture Notes in Computer Science, Springer-Verlag, 2009, pp. 137–151.