

AXIOMATIZING HYBRID XPATH WITH DATA

CARLOS ARECES AND RAUL FERVARI

Universidad Nacional de Córdoba & CONICET, Argentina

e-mail address: carlos.areces@unc.edu.ar

e-mail address: rfervari@unc.edu.ar

ABSTRACT. In this paper we introduce sound and strongly complete axiomatizations for XPath with data constraints extended with hybrid operators. First, we present $\text{HXPath}_=$, a multi-modal version of XPath with data, extended with nominals and the hybrid operator \textcircled{A} . Then, we introduce an axiomatic system for $\text{HXPath}_=$, and we prove it is strongly complete with respect to the class of abstract data models, i.e., data models in which data values are abstracted as equivalence relations. We prove a general completeness result similar to the one presented in, e.g., [BtC06], that ensures that certain extensions of the axiomatic system we introduce are also complete. The axiomatic systems that can be obtained in this way cover a large family of hybrid XPath languages over different classes of frames, for which we present concrete examples. In addition, we investigate axiomatizations over the class of tree models, structures widely used in practice. We show that a strongly complete, finitary, first-order axiomatization of hybrid XPath over trees does not exist, and we propose two alternatives to deal with this issue. We finally introduce filtrations to investigate the status of decidability of the satisfiability problem for these languages.

1. XPATH AS A MODAL LOGIC WITH DATA TESTS

XPath is, arguably, the most widely used query language for the eXtensible Markup Language (XML). Indeed, XPath is implemented in XSLT [Wad00] and XQuery [Wor02] and it is used in many specification and update languages (e.g., Saxon). It is, fundamentally, a general purpose language for addressing, searching, and matching pieces of an XML document. It is an open standard and constitutes a World Wide Web Consortium (W3C) Recommendation [CD99]. In [KRV15] XPath is adapted to be used as a powerful query language over knowledge bases. Core-XPath [GKP05] is the fragment of XPath 1.0 containing the navigational behaviour of XPath. It can express properties of the underlying tree structure of the XML document, such as the label (tag name) of a node, but it cannot express conditions on the actual data contained in the attributes. In other words, it is essentially a *classical modal logic* [BdRV01, BvBW06]. For instance the path expressions **child**, **parent** and **descendant-or-self** are basically the modal operators \diamond , \diamond^- and \diamond^* respectively.

Core-XPath has been well studied from a modal logic point of view. For instance, its satisfiability problem is known to be decidable even in the presence of Document Type Definitions (DTDs) [Mar04, BFG08]. Moreover, it is known that it is equivalent, in terms of

Key words and phrases: XPath, data graphs, axiomatizations, data trees, decidability.

expressive power, to first-order logic with two variables (FO2) over an appropriate signature on trees [Mdr05], and that it is strictly less expressive than Propositional Dynamic Logic (PDL) with converse over trees [BK08]. Sound and complete axiomatizations for Core-XPath have been introduced in [tCM09, tCLM10]. It has been argued that the study of XPath from a modal logic point of view is not merely a theoretical challenge but it has also concrete applications. For instance, by investigating the *expressive power* of XPath fragments it is possible to determine if an integrity constraint can be simplified. Also, query containment and query equivalence –two fundamental tasks in query optimization– can be reduced to the *satisfiability problem*. In particular these results have direct impact on, e.g., security [FCG04], type checking [MN07] and consistency of XML specifications [AFL02].

However, from a database perspective, Core-XPath falls short to define the most important construct in a query language: the *join*. Without the ability to relate nodes based on the actual data values of the attributes, the logic’s expressive power is inappropriate for many applications. The extension of Core-XPath with (in)equality tests between attributes of elements in an XML document is named Core-Data-XPath in [BMSS09]. Here, we will call this logic XPath₌. Models of XPath₌ are usually data trees which can be seen as XML documents. A data tree is a tree whose nodes contain a label from a finite alphabet and a data value from an infinite domain. From a modal logic perspective, these data trees are a particular class of *relational models*. In recent years other data structures have been considered, and XPath₌ has been used to query these structures. In particular, in this article we consider arbitrary *data graphs* as models for XPath₌. Data graphs are the underlying mathematical structure in *graph databases* (see, e.g., [LV12, RWE13, AG08]) and it is important to study the metalogical properties of languages to query this particular kind of models (see, e.g., [LMV16, ABFF18]). In this respect, we focus on a variant of XPath₌ which provides us with several interesting expressivity features.

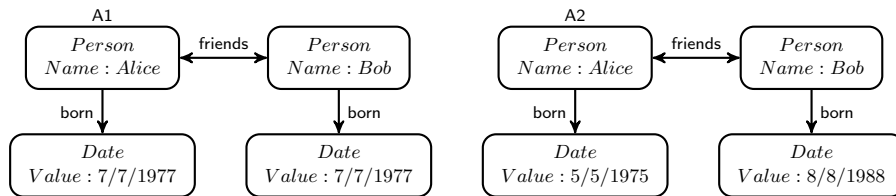


Figure 1: Example of a graph-structured database.

Figure 1 shows an example of a data graph, where we can see that edges carry labels from a finite alphabet (such as “friends”), and whose nodes carry data values, usually presented as a “*key:value*” pair, with *key* being a label from a finite alphabet (e.g., “Name”) and *value* being a data value from an infinite domain (e.g., “Alice”).

One of the characteristic advantages of graph databases is that they directly exhibit the *topology* of the data. Most of the approaches for querying these databases have focused on exploring the topology of the underlying graph exclusively. However, little attention has been paid to queries that check how the actual data contained in the graph nodes relates with the topology. XPath₌ allows comparison of data values by equality or inequality, even though it does not grant access to the concrete data values themselves.

The main characteristic of XPath₌ is to allow formulas of the form $\langle \alpha = \beta \rangle$ and $\langle \alpha \neq \beta \rangle$, where α, β are path expressions that navigate the graph using accessibility relations:

successor, predecessor, and for the particular case of tree-like models, descendant, next-sibling, etc.; and can make tests in intermediate nodes. In addition, in this work we consider the novel ‘@’ navigation operator that allows us to jump to named nodes. The formula $\langle \alpha = \beta \rangle$ (respectively $\langle \alpha \neq \beta \rangle$) is true at a node x of a data graph if there are nodes y, z that can be reached by paths satisfying the expressions α and β respectively, and such that the data value of y is equal (respectively different) to the data value of z .

The benefits of having data comparisons is described in the following example. Consider a social network which stores the information about the friendship relation between persons, and also stores the date of birth of each individual in the network. This scene can be represented as in Figure 1. Topological information is not sufficient to distinguish the Alice on the left, A1 from the Alice on the right, A2 (formally, these two nodes are bisimilar for the basic modal logic). However, the expression ‘the person that is friend of someone born the same day’ expressible in $\text{XPath}_=$, is true only at A1.

Recent articles investigate $\text{XPath}_=$ from a modal perspective. For example, satisfiability and evaluation are discussed in [Fig10, FS11, Fig12], while model theory and expressivity are studied in [ADF14, FFA14, FFA15, ADF17, ABFF16, ABFF18]. Query power and complexity of several $\text{XPath}_=$ fragments are investigated in [LMV16, Fig18]. In [BLS16], a Gentzen-style sequent calculus is given for a very restricted fragment of $\text{XPath}_=$, named DataGL. In DataGL, data comparisons are allowed only between the evaluation point and its successors. An extension of the equational axiomatic system from [tCLM10] is introduced in [ADFF17], allowing downward navigation and equality/inequality tests.

In this article, we will focus on the proof theory of $\text{XPath}_=$ extended with hybrid operators. It has been argued that the inclusion of hybrid operators in a modal language leads to a better proof theory [Bra11]. In particular, it can lead to general completeness results for axiomatizations by enabling Henkin style completeness proofs [BdRV01, BtC06]. Moreover, we will focus on strong completeness results (i.e., the axiomatizations we introduce characterize the consequence relation $\Gamma \vdash \varphi$ and not only theoremhood $\vdash \varphi$).

First results using hybridization techniques were introduced in [AF16] for a basic axiomatic system for $\text{XPath}_=$. In [AFS17] nominals are used as labels in a tableaux system.

Contribution. We will introduce a Hilbert-style axiomatization for $\text{XPath}_=$ with forward navigation for multiple accessibility relations, multiple equality/inequality comparisons, and where node expressions are extended with nominals (special labels that are valid in only one node), and path expressions are extended with the hybrid operator @ (allowing the navigation to some particular named node). We call this logic *Hybrid XPath with Data* (denoted $\text{HXPath}_=$). The language we consider is a variant of XPath fragments for data graphs in the literature. It improves the expressive power of the fragment $\text{XPath}_=$ from [ABFF16, ABFF18], e.g., by enabling us to express global properties. On the other hand, it has similarities with the fragment $\text{GXPath}_{\text{core}}(\text{eq})$ from [LMV16]. $\text{GXPath}_{\text{core}}(\text{eq})$ allows us to express data properties of nodes that are reachable in a finite number of steps, due to the reflexive-transitive navigation on atomic steps. However, the hybrid operator @ in $\text{HXPath}_=$ is enough to capture some of its expressivity by allowing us to ‘jump’ to nodes that are not accessible in one step, or that are simply not connected with the current node. One of the advantages of our approach, is that hybrid operators add only first-order behaviour, whereas reflexive-transitive closure needs second-order expressivity.

We will take advantage of hybrid operators to prove completeness using a Henkin-style model construction as presented in [Gol84, BdRV01, BtC06, SP10]. In fact, the use of the

so-called unorthodox inference rules (i.e., inference rules involving side conditions) will help us define an axiom system whose completeness can be automatically extended to more expressive logics.

Summing up, our contributions in this article are the following:

- We define a sound and strongly complete axiomatic system for multi-modal $\text{HXPath}_=$, refining the axioms and rules first presented in [AF16].
- We give a Henkin-style completeness proof, proving that the system is strongly complete for any extension given by pure axioms (formulas involving only nominals) and existential rules (instantiations of first-order formulas with quantification pattern $\forall\exists$ followed by an $\text{HXPath}_=$ -formula). This gives us a strong completeness result with respect to a wide variety of frame classes.
- Even though the main focus of our work is axiomatizing node expressions, we also provide as a by-product strong completeness results for path expressions.
- We discuss concrete extensions of $\text{HXPath}_=$ in which completeness is automatically obtained. In particular we explore backward and sibling navigation, and equality inclusion.
- We show that a strongly complete, finitary, first-order axiomatization of hybrid XPath over trees does not exist. Then we introduce an infinite pure axiomatic system that is strongly complete for a slightly larger class of tree models. We also prove that this system is weakly complete over tree models. As a corollary of the method we use, we conclude that the satisfiability problem for $\text{HXPath}_=$ on trees is decidable in EXPSpace .
- Finally, we prove that the satisfiability problem for the multi-modal logic $\text{HXPath}_=$ and some of its pure extensions is decidable in NEXPTIME , using filtrations to establish a bounded finite model property. To our knowledge, the best known lower bound can be obtained by PSPACE -hardness of mono-modal hybrid XPath with forward navigation [AFS17].

Related Work. It is argued in [BtC06] that hybrid logic is a natural setting for deductive systems involving Kripke semantics. By having nominals in the language, it is possible to imitate the role of first-order constants to obtain a Henkin-style completeness proof. Moreover, hybrid operators provide all the necessary machinery to get general completeness results for a wide variety of systems. The claim is that allowing the use of *unorthodox rules*, we can define a basic axiomatic system that can be extended with *pure axioms* and *existential rules*, which lets us obtain complete extensions with respect to diverse classes of frames, automatically.

One of the first proof systems for data-aware fragments of XPath was introduced in [BLS16]. It is a Gentzen-style system for a very restricted language named *DataGL*, interpreted on finite data trees. In *DataGL* formulas of the form $\diamond_{=}\varphi$ are read as: “the current node has the same data as a descendant where φ holds”. They introduced a sound and weakly complete sequent calculus for *DataGL* and established PSPACE -completeness for its validity problem.

An equational system which extends the one presented for navigational fragments of XPath in [tCLM10] was introduced in [ADFF17]. The authors consider first the fragment of $\text{XPath}_=$ without inequality tests, over data trees. Therein, the proof of weak completeness relies on a normal form theorem for both node and path expressions, and on a canonical model construction for consistent formulas in normal form inspired by [Fin75]. When

inequality is also taken into account, similar ideas can be used but the canonical model construction and the corresponding completeness proof become quite complex.

A Hilbert-style axiomatization for XPath₌ extended with hybrid operators and backward navigation was first introduced in [AF16], using ideas from hybrid logic: nominals play the role of first-order constants in a Henkin-style completeness proof. The system we present herein is a vast extension of the work from [AF16], and it automatically encompasses a large family of logics.

In [AFS17] nominals are used as labels in a tableaux system for a restricted variant of HXPath₌ with only one accessibility relation and one equality relation. The tableaux calculus is shown to be sound, complete and terminating, and it is used to prove that the satisfiability problem for such logic is PSPACE-complete.

Organization. The article is organized as follows. In Section 2 we introduce the syntax, semantics and a notion of bisimulation for HXPath₌. We define the axiomatic system HXP in Section 3 and we prove its completeness. We finish that section by introducing extensions of HXP and their complete axiomatizations with pure axioms and existential rules. In Section 4 we introduce an axiomatic system for XPath over the class of data trees. Then, we use filtrations to show that the satisfiability problem of HXPath₌ is decidable in Section 5. To conclude, in Section 6 we discuss the results and introduce future lines of research.

2. HYBRID XPATH WITH DATA

In this section we introduce the syntax and semantics for the logic we call Hybrid XPath with Data (HXPath₌ for short). Then, we present a notion of bisimulation for HXPath₌.

2.1. Syntax and Semantics of HXPath₌. Throughout the text, let **Prop** be a countable set of propositional symbols; let **Nom** be a countable set of nominals such that $\text{Prop} \cap \text{Nom} = \emptyset$; and let **Mod** and **Eq** be sets of modal and equality symbols, respectively.

Definition 2.1 (Syntax). The sets PExp of path expressions (which we will note as $\alpha, \beta, \gamma, \dots$) and NExp of node expressions (which we will note as $\varphi, \psi, \theta, \dots$) of HXPath₌ are defined by mutual recursion as follows:

$$\begin{aligned} \text{PExp} &::= a \mid @_i \mid [\varphi] \mid \alpha\beta \\ \text{NExp} &::= p \mid i \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle \alpha =_e \beta \rangle \mid \langle \alpha \neq_e \beta \rangle, \end{aligned}$$

where $p \in \text{Prop}$, $i \in \text{Nom}$, $a \in \text{Mod}$, $e \in \text{Eq}$, $\varphi, \psi \in \text{NExp}$, and $\alpha, \beta \in \text{PExp}$. We will refer to members of $\text{PExp} \cup \text{NExp}$ as *expressions*. In what follows, when referring to expressions of HXPath₌ we will reserve the term *formula* for members of NExp. We say that an expression is *pure* if it does not contain propositional symbols.

Notice that path expressions occur in node expressions in *data comparisons* of the form $\langle \alpha =_e \beta \rangle$ and $\langle \alpha \neq_e \beta \rangle$, while node expressions occur in path expressions in *tests* like $[\varphi]$.

In what follows we will always use $*$ for $=_e$ and \neq_e , when the particular operator used is not relevant. Other Boolean operators are defined as usual: $\varphi \vee \psi := \neg(\neg\varphi \wedge \neg\psi)$, and $\varphi \rightarrow \psi := \neg\varphi \vee \psi$. Below we define other operators as abbreviations.

Definition 2.2 (Abbreviations). Let α, β, δ be path expressions, γ_1, γ_2 path expressions or the empty string, φ a node expression, i a nominal, and p an arbitrary symbol in **Prop**. We define the following expressions:

$$\begin{aligned}
\top &:= p \vee \neg p \\
\perp &:= \neg \top \\
\epsilon &:= [\top] \\
[\alpha =_e \beta] &:= \neg \langle \alpha \neq_e \beta \rangle \\
[\alpha \neq_e \beta] &:= \neg \langle \alpha =_e \beta \rangle \\
\langle \alpha \rangle \varphi &:= \langle \alpha[\varphi] =_e \alpha[\varphi] \rangle \\
[\alpha] \varphi &:= \neg \langle \alpha \rangle \neg \varphi \\
@_i \varphi &:= \langle @_i \rangle \varphi \\
\langle \gamma_1(\alpha \cup \beta) \gamma_2 * \delta \rangle &:= \langle \gamma_1 \alpha \gamma_2 * \delta \rangle \vee \langle \gamma_1 \beta \gamma_2 * \delta \rangle \\
\langle \delta * \gamma_1(\alpha \cup \beta) \gamma_2 \rangle &:= \langle \delta * \gamma_1 \alpha \gamma_2 \rangle \vee \langle \delta * \gamma_1 \beta \gamma_2 \rangle
\end{aligned}$$

As a corollary of the definition of the semantic relation below, the diamond and box expressions $\langle \alpha \rangle \varphi$ and $[\alpha] \varphi$ will have their classical meaning, and the same will be true for hybrid formulas of the form $@_i \varphi$. Notice that we use $@_i$ both as a path expression and as a modality; the intended meaning will always be clear in context. Notice also that, following the standard notation in XPath logics and in modal logics, the $[\]$ operation is overloaded: for φ a node expression and α a path expression, both $[\alpha] \varphi$ and $\alpha[\varphi]$ are well-formed expressions; the former is a node expression where $[\alpha]$ is a box modality, the latter is a path expression where $[\varphi]$ is a test.

We now define the structures that will be used to evaluate formulas in the language.

Definition 2.3 (Hybrid Data Models). An *abstract hybrid data model* is a tuple $\mathcal{M} = \langle M, \{\sim_e\}_{e \in \text{Eq}}, \{R_a\}_{a \in \text{Mod}}, V, \text{nom} \rangle$, where M is a non-empty set of elements; for each $e \in \text{Eq}$, $\sim_e \subseteq M \times M$ is an equivalence relation between elements of M ; for each $a \in \text{Mod}$, $R_a \subseteq M \times M$ is the associated accessibility relation; $V : M \rightarrow 2^{\text{Prop}}$ is the valuation function; and $\text{nom} : \text{Nom} \rightarrow M$ is a function that assigns nominals to certain elements. Given $m \in M$, (\mathcal{M}, m) is called a *pointed model* (parentheses are usually dropped).

An *abstract data frame* (or just *frame*) is a tuple $\mathcal{F} = \langle M, \{\sim_e\}_{e \in \text{Eq}}, \{R_a\}_{a \in \text{Mod}} \rangle$. Let \mathfrak{F} be a class of frames, its *class of models* is $\text{Mod}(\mathfrak{F}) = \{ \langle M, \{\sim_e\}_{e \in \text{Eq}}, \{R_a\}_{a \in \text{Mod}}, V, \text{nom} \rangle \mid \langle M, \{\sim_e\}_{e \in \text{Eq}}, \{R_a\}_{a \in \text{Mod}} \rangle \in \mathfrak{F}, \text{ for } V, \text{nom arbitrary} \}$.

A *concrete hybrid data model* is a tuple $\mathcal{M} = \langle M, D, \{R_a\}_{a \in \text{Mod}}, V, \text{nom}, \text{data} \rangle$, where M is a non-empty set of elements; D is a non-empty set of data; for each $a \in \text{Mod}$, $R_a \subseteq M \times M$ is the associated accessibility relation; $V : M \rightarrow 2^{\text{Prop}}$ is the valuation function; $\text{nom} : \text{Nom} \rightarrow M$ is a function which names some nodes; and $\text{data} : \text{Eq} \times M \rightarrow D$ is a function which assigns a data value to each node of the model (for each type of equality considered in the model).

Concrete data models are most commonly used in applications, where we encounter data from an infinite alphabet (e.g., alphabetic strings) associated to the nodes in a semi-structured database, and different ways of comparing this data. It is easy to see that to each concrete data model we can associate an abstract data model where data is replaced by an equivalence relation that links all nodes with the same data, for each equality symbol from Eq . Vice-versa, each abstract data model can be “concretized” by assigning to each pair node and symbol from Eq , its equivalence data class as data. We will prove soundness and completeness over the class of abstract data models and, as a corollary, obtain completeness over concrete data models.

Definition 2.4 (Semantics). Let $\mathcal{M} = \langle M, \{\sim_e\}_{e \in \text{Eq}}, \{R_a\}_{a \in \text{Mod}}, V, \text{nom} \rangle$ be an abstract data model, and $m, n \in M$. We define the semantics of $\text{HXPath}_=$ as follows:

$$\begin{aligned}
\mathcal{M}, m, n \models \mathbf{a} & \text{ iff } mR_a n \\
\mathcal{M}, m, n \models @_i & \text{ iff } \text{nom}(i) = n \\
\mathcal{M}, m, n \models [\varphi] & \text{ iff } m = n \text{ and } \mathcal{M}, m \models \varphi \\
\mathcal{M}, m, n \models \alpha\beta & \text{ iff there is } l \in M \text{ s.t. } \mathcal{M}, m, l \models \alpha \text{ and } \mathcal{M}, l, n \models \beta \\
\mathcal{M}, m \models p & \text{ iff } p \in V(m) \\
\mathcal{M}, m \models i & \text{ iff } \text{nom}(i) = m \\
\mathcal{M}, m \models \neg\varphi & \text{ iff } \mathcal{M}, m \not\models \varphi \\
\mathcal{M}, m \models \varphi \wedge \psi & \text{ iff } \mathcal{M}, m \models \varphi \text{ and } \mathcal{M}, m \models \psi \\
\mathcal{M}, m \models \langle \alpha =_e \beta \rangle & \text{ iff there are } n, l \in M \text{ s.t. } \mathcal{M}, m, n \models \alpha, \mathcal{M}, m, l \models \beta \text{ and } n \sim_e l \\
\mathcal{M}, m \models \langle \alpha \neq_e \beta \rangle & \text{ iff there are } n, l \in M \text{ s.t. } \mathcal{M}, m, n \models \alpha, \mathcal{M}, m, l \models \beta \text{ and } n \not\sim_e l.
\end{aligned}$$

We say a node expression φ (path expression α) is *satisfiable* if there exists \mathcal{M}, m (\mathcal{M}, m, n) such that $\mathcal{M}, m \models \varphi$ ($\mathcal{M}, m, n \models \alpha$). Satisfiability for sets of node and path expressions is defined in the obvious way. Let $\Gamma \cup \{\varphi\}$ be a set of node expressions, $\Gamma \models \varphi$ if for all \mathcal{M}, m , $\mathcal{M}, m \models \Gamma$ implies $\mathcal{M}, m \models \varphi$. For a class \mathfrak{C} of models, $\Gamma \models_{\mathfrak{C}} \varphi$ if for all $\mathcal{M} \in \mathfrak{C}$ and m in the domain of \mathcal{M} , $\mathcal{M}, m \models \Gamma$ implies $\mathcal{M}, m \models \varphi$. When $\Gamma = \emptyset$ we write $\models \varphi$ (respectively $\models_{\mathfrak{C}} \varphi$).

As mentioned, it is a straightforward exercise to show that modal and hybrid operators have their intended meaning.

Proposition 2.5. *Let \mathcal{M} be an abstract hybrid data model and m a state in \mathcal{M} . Then*

$$\begin{aligned}
\mathcal{M}, m \models @_i \varphi & \text{ iff } \mathcal{M}, \text{nom}(i) \models \varphi \\
\mathcal{M}, m \models \langle \mathbf{a} \rangle \varphi & \text{ iff for some } n \in M \text{ s.t. } mR_a n, \mathcal{M}, n \models \varphi \\
\mathcal{M}, m \models [\mathbf{a}] \varphi & \text{ iff for all } n \in M, mR_a n \text{ implies } \mathcal{M}, n \models \varphi.
\end{aligned}$$

The addition of the hybrid operators to XPath increases its expressive power. The following examples should serve as illustration.

Example 2.6. We list below some $\text{HXPath}_=$ expressions together with their intuitive meaning:

$\alpha[i]$	There exists an α path between the current points of evaluation; the second node is named i .
$@_i \alpha$	There exists an α path between the node named i and some other node.
$\langle @_i =_e @_j \rangle$	The node named i has the same data as the node named j , w.r.t. the data field e .
$\langle \alpha =_e @_i \beta \rangle$	There exists a node accessible from the current point of evaluation by an α path that has the same data than a node accessible from the point named i by a β path, w.r.t. the data field e .

The next example illustrates the expressivity gained by adding hybrid operators into the language, in a concrete example.

Example 2.7. Consider the following queries in the data graph from Figure 1 (we omit the subindex in the symbols $=$ and \neq , since we deal with only one equality relation).

$\langle @_{A1} \text{born}[Value] = @_{A1} \text{friends born}[Value] \rangle$	The person with the id A1 is friend of someone born in the same day.
$[@_{A2} \text{born}[Value] \neq @_{A2} \text{friends born}[Value]]$	All friends of the person with id A2 were born in a day different to hers.
$\langle @_{A1}[Name] = @_{A2}[Name] \rangle$ $\wedge \langle @_{A1} \text{born}[Value] \neq @_{A2} \text{born}[Value] \rangle$	The persons with id A1 and A2 have the same value (both are named Alice) and they were born in different days.

In the last item, the conjunct $\langle @_{A1}[Name] = @_{A2}[Name] \rangle$ states that the nodes named by the nominals A1 and A2 contain a data value corresponding to the key *Name*, expressed by the test $[Name]$; also, these data values coincide, without referring to the actual value.

It is worth noting that with classical XPath₌ navigation, the two first properties can be expressed only if we are evaluating the formulas in the node with id A1 and A2, respectively. This is due the fact that XPath₌ only allows local navigation. Moreover, the third formula is not expressible in XPath₌, even in the presence of transitive closure operators, as it involves unconnected components of the graph.

Before concluding this section, we will introduce two classical notions that will be useful in the rest of the paper: the *modal depth* and the set of *subformulas* of a formula. These definitions are given on node and path expressions seen as strings, in particular they consider the empty string λ which is not in the language. However, these definitions work as intended.

Definition 2.8. We define the *modal depth* of an expression, by mutual recursion as follows.

$$\begin{array}{ll}
\text{md}(\lambda) &= 0 & \text{md}(p) &= 0 \\
\text{md}(a\alpha) &= 1 + \text{md}(\alpha) & \text{md}(\neg\varphi) &= \text{md}(\varphi) \\
\text{md}([\varphi]\alpha) &= \max(\text{md}(\varphi), \text{md}(\alpha)) & \text{md}(\varphi \wedge \psi) &= \max(\text{md}(\varphi), \text{md}(\psi)) \\
\text{md}(@_i\alpha) &= \text{md}(\alpha) & \text{md}(\langle \alpha * \beta \rangle) &= \max(\text{md}(\alpha), \text{md}(\beta)),
\end{array}$$

where λ represents the empty string, $a \in \text{Mod}$, $p \in \text{Prop} \cup \text{Nom}$, $i \in \text{Nom}$, $* \in \{=_e, \neq_e\}$, $\varphi, \psi \in \text{NExp}$ and $\alpha, \beta \in \text{PExp}$.

Definition 2.9. We define the set of *subformulas* of an expression, by mutual recursion as follows.

$$\begin{array}{ll}
\text{Sub}(\lambda) &= \emptyset & \text{Sub}(p) &= \{p\} \\
\text{Sub}(a\alpha) &= \{\langle a\alpha \rangle \top\} \cup \text{Sub}(\alpha) & \text{Sub}(\neg\varphi) &= \{\neg\varphi\} \cup \text{Sub}(\varphi) \\
\text{Sub}([\varphi]\alpha) &= \{\langle [\varphi]\alpha \rangle \top\} \cup \text{Sub}(\varphi) \cup \text{Sub}(\alpha) & \text{Sub}(\varphi \wedge \psi) &= \{\varphi \wedge \psi\} \cup \text{Sub}(\varphi) \cup \text{Sub}(\psi) \\
\text{Sub}(@_i\alpha) &= \{\langle @_i\alpha \rangle \top, i\} \cup \text{Sub}(\alpha) & \text{Sub}(\langle \alpha * \beta \rangle) &= \{\langle \alpha * \beta \rangle\} \cup \text{Sub}(\alpha) \cup \text{Sub}(\beta),
\end{array}$$

where λ represents the empty string, $a \in \text{Mod}$, $p \in \text{Prop} \cup \text{Nom}$, $i \in \text{Nom}$, $* \in \{=_e, \neq_e\}$, $\varphi, \psi \in \text{NExp}$ and $\alpha, \beta \in \text{PExp}$.

2.2. A Note on Bisimulations. In the rest of the paper we will sometimes need the notion of bisimulation for HXPath₌. Bisimulation for XPath on data trees was investigated in [FFA14, FFA15]. In [ABFF16, ABFF18] it is shown that the same notion can be used in arbitrary models. It is simple to extend these definitions to take into account hybrid operators (see, e.g., [ABM01, BDRV01]).

Definition 2.10 (Bisimulations). Let $\mathcal{M} = \langle M, \{\sim_e\}_{e \in \text{Eq}}, \{R_a\}_{a \in \text{Mod}}, V, \text{nom} \rangle$ and $\mathcal{M}' = \langle M', \{\sim'_e\}_{e \in \text{Eq}}, \{R'_a\}_{a \in \text{Mod}}, V', \text{nom}' \rangle$ be two abstract hybrid data models, $m \in M$, $m' \in M'$. An *HXPath₌-bisimulation* between \mathcal{M}, m and \mathcal{M}', m' is a relation $Z \subseteq M \times M'$ such that mZm' , and when lZl' we have:

- **(Harmony)** $V(l) = V'(l')$ and $l = \text{nom}(i)$ iff $l' = \text{nom}'(i)$ (for all $i \in \text{Nom}$).
- **(Zig₌)** If there are paths $\pi_1 = lR_{a_1}h_1R_{a_2} \dots R_{a_{n_1}}h_{n_1}$ and $\pi_2 = lR_{b_1}k_1R_{b_2} \dots R_{b_{n_2}}k_{n_2}$ (for $a_i, b_i \in \text{Mod}$), then there are paths $\pi'_1 = l'R'_{a_1}h'_1R'_{a_2} \dots R'_{a_{n_1}}h'_{n_1}$ and $\pi'_2 = l'R'_{b_1}k'_1R'_{b_2} \dots R'_{b_{n_2}}k'_{n_2}$ such that
 - (1) $h_iZh'_i$ for $1 \leq i \leq n_1$.
 - (2) $k_iZk'_i$ for $1 \leq i \leq n_2$.
 - (3) $h_{n_1} \sim_e k_{n_2}$ iff $h'_{n_1} \sim'_e k'_{n_2}$, for all $e \in \text{Eq}$.
- **(Zag₌)** If there are paths $\pi'_1 = l'R'_{a_1}h'_1R'_{a_2} \dots R'_{a_{n_1}}h'_{n_1}$ and $\pi'_2 = l'R'_{b_1}k'_1R'_{b_2} \dots R'_{b_{n_2}}k'_{n_2}$ (for $a_i, b_i \in \text{Mod}$), then there are paths $\pi_1 = lR_{a_1}h_1R_{a_2} \dots R_{a_{n_1}}h_{n_1}$ and $\pi_2 = lR_{b_1}k_1R_{b_2} \dots R_{b_{n_2}}k_{n_2}$ such that conditions 1, 2 and 3 above are verified.
- **(Nom)** For all $i \in \text{Nom}$, $\text{nom}(i)Z\text{nom}'(i)$.

We write $\mathcal{M}, m \rightleftharpoons \mathcal{M}', m'$ (and say that \mathcal{M}, m and \mathcal{M}', m' are *HXPath₌-bisimilar*) if there is an *HXPath₌-bisimulation* Z such that mZm' .

The notion of *HXPath₌-bisimulation* extends the one for basic modal logic [BdRV01]. As usual, **(Harmony)** takes care of atomic information (both propositional symbols and nominals). The standard **(Zig)** and **(Zag)** should be strengthened to take into account that XPath modalities deal with two paths at the same time, and moreover equality of data values can be checked at their ending points. Finally **(Nom)** takes care of the @ operator.

It is straightforward to prove that *HXPath₌-bisimilar* models satisfy the same *HXPath₌-formulas* (see [ABFF18]).

Proposition 2.11. *Let \mathcal{M}, m and \mathcal{M}', m' be two pointed data models. If $\mathcal{M}, m \rightleftharpoons \mathcal{M}', m'$ then for any *HXPath₌-formula* φ , $\mathcal{M}, m \models \varphi$ if and only if $\mathcal{M}', m' \models \varphi$.*

We can extend the notion of ℓ -bisimulation introduced in [FFA15] in a similar way. An ℓ -bisimulation checks model equivalence up to a certain modal depth.

Definition 2.12 (ℓ -bisimulations). Let $\mathcal{M} = \langle M, \{\sim_e\}_{e \in \text{Eq}}, \{R_a\}_{a \in \text{Mod}}, V, \text{nom} \rangle$ and $\mathcal{M}' = \langle M', \{\sim'_e\}_{e \in \text{Eq}}, \{R'_a\}_{a \in \text{Mod}}, V', \text{nom}' \rangle$ be two abstract hybrid data models, $m \in M$, $m' \in M'$. We say that \mathcal{M}, m and \mathcal{M}', m' are ℓ -bisimilar for *HXPath₌* (written $\mathcal{M}, m \rightleftharpoons_\ell \mathcal{M}', m'$) if there exists a family of relations $(Z_j)_{j \leq \ell}$ in $M \times M'$ such that $mZ_\ell m'$, and for all $j \leq \ell$, when $lZ_j l'$ we have:

- **(Harmony)** $V(l) = V'(l')$ and $l = \text{nom}(i)$ iff $l' = \text{nom}'(i)$ (for all $i \in \text{Nom}$).
- **(Zig₌)** If there are paths $\pi_1 = lR_{a_1}h_1R_{a_2} \dots R_{a_{n_1}}h_{n_1}$ and $\pi_2 = lR_{b_1}k_1R_{b_2} \dots R_{b_{n_2}}k_{n_2}$ (for $a_i, b_i \in \text{Mod}$), $n_1, n_2 \leq j$, then there are paths $\pi'_1 = l'R'_{a_1}h'_1R'_{a_2} \dots R'_{a_{n_1}}h'_{n_1}$ and $\pi'_2 = l'R'_{b_1}k'_1R'_{b_2} \dots R'_{b_{n_2}}k'_{n_2}$ such that
 - (1) $h_iZ_{(j-n_1)+i}h'_i$ for $1 \leq i \leq n_1$.
 - (2) $k_iZ_{(j-n_2)+i}k'_i$ for $1 \leq i \leq n_2$.
 - (3) $h_{n_1} \sim_e k_{n_2}$ iff $h'_{n_1} \sim'_e k'_{n_2}$, for all $e \in \text{Eq}$.
- **(Zag₌)** If there are paths $\pi'_1 = l'R'_{a_1}h'_1R'_{a_2} \dots R'_{a_{n_1}}h'_{n_1}$ and $\pi'_2 = l'R'_{b_1}k'_1R'_{b_2} \dots R'_{b_{n_2}}k'_{n_2}$ (for $a_i, b_i \in \text{Mod}$), $n_1, n_2 \leq j$, then there are paths $\pi_1 = lR_{a_1}h_1R_{a_2} \dots R_{a_{n_1}}h_{n_1}$ and $\pi_2 = lR_{b_1}k_1R_{b_2} \dots R_{b_{n_2}}k_{n_2}$ such that conditions 1, 2 and 3 above are verified.

- **(Nom)** For all $i \in \text{Nom}$, $\text{nom}(i)Z_\ell \text{nom}'(i)$.

Again, it is straightforward to prove the next proposition.

Proposition 2.13. *Let \mathcal{M}, m and \mathcal{M}', m' be two pointed data models. If $\mathcal{M}, m \simeq_\ell \mathcal{M}', m'$ then for any $\text{HXPath}_=$ -formula φ such that $\text{md}(\varphi) \leq \ell$, $\mathcal{M}, m \models \varphi$ if and only if $\mathcal{M}', m' \models \varphi$.*

3. AXIOMATIZATION

In this section we introduce the axiomatic system HXP for $\text{HXPath}_=$. It is an extension of an axiomatic system for the hybrid logic $\mathcal{HL}(@)$ which adds nominals and the @ operator to the basic modal language (see [BdRV01]). In particular, we include axioms to handle data equality and inequality. We will prove that HXP is sound and strongly complete with respect to the class of abstract hybrid data models. Moreover, we will show that the system is strongly complete for any extension with some particular kind of axioms and inference rules. This result will be helpful in order to automatically obtain complete axiomatic systems for several natural extensions of the language $\text{HXPath}_=$. The system is geared for inference over node expressions, but we will discuss path expressions in Section 3.4. We remind the reader that when referring to $\text{HXPath}_=$ we use the term *formula* for node expressions. In what follows, we use $*$ in axioms which hold for both $=_e$ and \neq_e .

3.1. The Basic Axiomatic System HXP. We present axioms and rules step by step, providing brief comments to help the reader understand their role. In all axioms and rules φ, ψ and θ are node expressions; α, β, γ and η are path expressions; and i, j and k are nominals. More precisely, we provide *axiom and rule schemes*, i.e., they can be instantiated with arbitrary path and node expressions, respecting typing. Hence, φ, ψ and θ can be instantiated with node expressions; α, β, γ and η with path expressions; and i, j and k with nominals.

Definition 3.1 (Theorems, Syntactic Consequence, Consistency). For \mathbf{A} an axiomatic system, φ is a *theorem of \mathbf{A}* (notation: $\vdash_{\mathbf{A}} \varphi$) if it is either an instantiation of an axiom of \mathbf{A} , or it can be derived from an axiom of \mathbf{A} in a finite number of steps by application of the rules of \mathbf{A} . Let Γ be a set of node expressions. We write $\Gamma \vdash_{\mathbf{A}} \varphi$ and say that φ is a *syntactic consequence of Γ in \mathbf{A}* if there exists a finite set $\Gamma' \subseteq \Gamma$ such that $\vdash_{\mathbf{A}} \bigwedge \Gamma' \rightarrow \varphi$ (where $\bigwedge \emptyset = \top$). We say that Γ is *consistent (for \mathbf{A})* if $\Gamma \not\vdash_{\mathbf{A}} \perp$. We write \vdash instead of $\vdash_{\mathbf{A}}$ if \mathbf{A} is clear from context.

In addition to a complete set of axioms and rules for propositional logic, HXP includes generalizations of the K axiom and the *Necessitation* rule for the basic modal logic to handle modalities with arbitrary path expressions.

Axiom and rule for classical modal logic

$$K \quad \vdash [\alpha](\varphi \rightarrow \psi) \rightarrow ([\alpha]\varphi \rightarrow [\alpha]\psi) \qquad \frac{\vdash \varphi}{\vdash [\alpha]\varphi} \text{Nec}$$

Then we introduce generalizations of the inference rules for the hybrid logic $\mathcal{HL}(@)$.

Hybrid rules

$$\frac{\vdash @_j \varphi}{\vdash \varphi} \textit{name} \qquad \frac{\vdash @_i \langle \mathbf{a} \rangle j \wedge \langle @_j \alpha * \beta \rangle \rightarrow \theta}{\vdash \langle @_i \mathbf{a} \alpha * \beta \rangle \rightarrow \theta} \textit{paste}$$

j is a nominal different from i that does not occur in $\varphi, \alpha, \beta, \theta$.

Now we introduce axioms to handle $@$. Notice that $@_i$ is a path expression and as a result, some of the standard hybrid axioms for $@$ have been generalized. In particular, the K axiom and Nec rule above also apply to $@_i$. In addition, we provide axioms to ensure that the relation induced by $@$ is a congruence.

Axioms for $@$	Congruence for $@$
$@$ -self-dual $\vdash \neg @_i \varphi \leftrightarrow @_i \neg \varphi$	$@$ -refl $\vdash @_i i$
$@$ -intro $\vdash i \rightarrow (\varphi \leftrightarrow @_i \varphi)$	agree $\vdash \langle @_j @_i \alpha * \beta \rangle \leftrightarrow \langle @_i \alpha * \beta \rangle$
	back $\vdash \langle \gamma @_i \alpha * \beta \rangle \rightarrow \langle @_i \alpha * \beta \rangle$

Axioms involving the classical XPath operators can be found below. First we introduce axioms to handle complex path expressions in data comparisons. Then we introduce axioms to handle data tests.

Axioms for paths

$$\begin{aligned} \textit{comp-assoc} & \vdash \langle (\alpha\beta)\gamma * \eta \rangle \leftrightarrow \langle \alpha(\beta\gamma) * \eta \rangle \\ \textit{comp-neutral} & \vdash \langle \alpha\beta * \gamma \rangle \leftrightarrow \langle \alpha\epsilon\beta * \gamma \rangle \text{ (}\alpha \text{ or } \beta \text{ can be empty)} \\ \textit{comp-dist} & \vdash \langle \alpha\beta \rangle \varphi \leftrightarrow \langle \alpha \rangle \langle \beta \rangle \varphi \end{aligned}$$

Axioms for data

$$\begin{aligned} \textit{equal} & \vdash \langle \epsilon =_e \epsilon \rangle \\ \textit{distinct} & \vdash \neg \langle \epsilon \neq_e \epsilon \rangle \\ @\textit{-data} & \vdash \neg \langle @_i =_e @_j \rangle \leftrightarrow \langle @_i \neq_e @_j \rangle \\ \epsilon\textit{-trans} & \vdash \langle \epsilon =_e \alpha \rangle \wedge \langle \epsilon =_e \beta \rangle \rightarrow \langle \alpha =_e \beta \rangle \\ *\textit{-comm} & \vdash \langle \alpha * \beta \rangle \leftrightarrow \langle \beta * \alpha \rangle \\ *\textit{-test} & \vdash \langle [\varphi] \alpha * \beta \rangle \leftrightarrow \varphi \wedge \langle \alpha * \beta \rangle \\ @*\textit{-dist} & \vdash \langle @_i \alpha * @_i \beta \rangle \rightarrow @_i \langle \alpha * \beta \rangle \\ \textit{subpath} & \vdash \langle \alpha\beta * \gamma \rangle \rightarrow \langle \alpha \rangle \top \\ \textit{comp*}\textit{-dist} & \vdash \langle \alpha \rangle \langle \beta * \gamma \rangle \rightarrow \langle \alpha\beta * \alpha\gamma \rangle \end{aligned}$$

It is a straightforward exercise to see that the axiomatic system HXP is sound. However, we will provide a more general statement of soundness later on.

Below we prove that some useful theorems and rules can be derived within HXP.

Proposition 3.2. *The following are theorems and derived rules of HXP, and will be used (explicitly or implicitly) in the rest of the paper.*

$test-dist$	$\vdash \langle [\varphi] =_e [\psi] \rangle \leftrightarrow \varphi \wedge \psi$	
$test-\perp$	$\vdash \langle [\varphi] \neq_e [\psi] \rangle \leftrightarrow \perp$	
$@-swap$	$\vdash @_i \langle \alpha * @_j \beta \rangle \leftrightarrow @_j \langle \beta * @_i \alpha \rangle$	
$@-intro'$	$\vdash (i \wedge \varphi) \rightarrow @_i \varphi$	
$@-sym.$	$\vdash @_i j \rightarrow @_j i$	
$K_{@}^{-1}$	$\vdash (@_i \varphi \rightarrow @_i \psi) \rightarrow @_i (\varphi \rightarrow \psi)$	
nom	$\vdash @_i j \wedge @_i \alpha * \beta \rightarrow @_j \alpha * \beta$	
$bridge$	$\vdash \langle \alpha \rangle i \wedge @_i \varphi \rightarrow \langle \alpha \rangle \varphi$	
$name'$	$If \vdash i \rightarrow \varphi$ then $\vdash \varphi$, if i does not appear in φ	

Proof. ($test-dist$ and $test-\perp$). Let $*$ be $=_e$ or \neq_e . Then:

$\vdash \langle [\varphi] * [\psi] \rangle \leftrightarrow \langle [\varphi] \epsilon * [\psi] \rangle$	$(comp-neutral)$
$\vdash \langle [\varphi] \epsilon * [\psi] \rangle \leftrightarrow \varphi \wedge \langle \epsilon * [\psi] \rangle$	$(*test)$
$\vdash \varphi \wedge \langle \epsilon * [\psi] \rangle \leftrightarrow \varphi \wedge \langle [\psi] * \epsilon \rangle$	$(*comm)$
$\vdash \varphi \wedge \langle [\psi] * \epsilon \rangle \leftrightarrow \varphi \wedge \langle [\psi] \epsilon * \epsilon \rangle$	$(comp-neutral)$
$\vdash \varphi \wedge \langle [\psi] \epsilon * \epsilon \rangle \leftrightarrow \varphi \wedge \psi \wedge \langle \epsilon * \epsilon \rangle$	$(*test)$

Replacing $*$ by $=_e$ we get $\varphi \wedge \psi$ by *equal*. Replacing it by \neq_e we get $\langle \epsilon \neq_e \epsilon \rangle$, and given *distinct* we can derive \perp .

$(@-swap)$.

$\vdash @_i \langle \alpha =_e @_j \beta \rangle \leftrightarrow \langle @_i \alpha =_e @_i @_j \beta \rangle$	$(@=-dist)$
$\vdash \langle @_i \alpha =_e @_i @_j \beta \rangle \leftrightarrow \langle @_i @_j \beta =_e @_i \alpha \rangle$	$(=-comm)$
$\vdash \langle @_i @_j \beta =_e @_i \alpha \rangle \leftrightarrow \langle @_j \beta =_e @_i \alpha \rangle$	$(agree)$
$\vdash \langle @_j \beta =_e @_i \alpha \rangle \leftrightarrow \langle @_i \alpha =_e @_j \beta \rangle$	$(=-comm)$
$\vdash \langle @_i \alpha =_e @_j \beta \rangle \leftrightarrow \langle @_j @_i \alpha =_e @_j \beta \rangle$	$(agree)$
$\vdash \langle @_j @_i \alpha =_e @_j \beta \rangle \leftrightarrow @_j \langle @_i \alpha =_e \beta \rangle$	$(agree)$
$\vdash @_j \langle @_i \alpha =_e \beta \rangle \leftrightarrow @_j \langle \beta =_e @_i \alpha \rangle$	$(=-comm)$

$(@-intro')$. Direct from $@-intro$.

$(@-sym.)$.

$\vdash (j \wedge i) \rightarrow @_j i$	$(@-intro')$
$\vdash (@_i j \wedge @_i i) \rightarrow @_i @_j i$	(Nec, K)
$\vdash @_i j \rightarrow @_j i$	$(@-refl., back)$

$(K_{@}^{-1})$.

$\vdash (@_i \neg(\varphi \rightarrow \psi) \rightarrow @_i \varphi) \wedge (@_i \neg(\varphi \rightarrow \psi) \rightarrow @_i \neg\psi)$	$(prop, Nec, K)$
$\vdash @_i \neg(\varphi \rightarrow \psi) \rightarrow (@_i \varphi \wedge @_i \neg\psi)$	$(prop)$
$\vdash \neg @_i (\varphi \rightarrow \psi) \rightarrow (@_i \varphi \wedge \neg @_i \psi)$	$(@-self-dual)$
$\vdash (@_i \varphi \rightarrow @_i \psi) \rightarrow @_i (\varphi \rightarrow \psi)$	$(prop)$

(nom) .

$\vdash (j \wedge @_j \beta * \alpha) \rightarrow @_j \langle @_j \beta * \alpha \rangle$	$(@-intro')$
$\vdash (@_i j \wedge @_i \langle @_j \beta * \alpha \rangle) \rightarrow @_i @_j \langle @_j \beta * \alpha \rangle$	(Nec, K)

$$\begin{aligned}
\vdash (\@_i j \wedge \@_j \langle \@_i \alpha * \beta \rangle) &\rightarrow \@_i \@_j \langle \@_j \alpha * \beta \rangle && (\text{@-swap, *-comm}) \\
\vdash (\@_j \@_i j \wedge \@_j \langle \@_i \alpha * \beta \rangle) &\rightarrow \@_j \langle \@_j \alpha * \beta \rangle && (\text{agree}) \\
\vdash \@_j (\@_i j \wedge \langle \@_i \alpha * \beta \rangle) &\rightarrow \langle \@_j \alpha * \beta \rangle && (K_{\text{@}}^{-1}) \\
\vdash \@_i j \wedge \langle \@_i \alpha * \beta \rangle &\rightarrow \langle \@_j \alpha * \beta \rangle && (\text{name})
\end{aligned}$$

(*bridge*). Using contrapositive, *bridge* is equivalent to $\langle \alpha \rangle i \wedge [\alpha] \varphi \rightarrow \@_i \varphi$. Using the modal theorem $\vdash \langle \alpha \rangle \varphi \wedge [\alpha] \psi \rightarrow \langle \alpha \rangle (\varphi \wedge \psi)$, we reason:

$$\begin{aligned}
&\vdash \langle \alpha \rangle i \wedge [\alpha] \varphi \rightarrow \langle \alpha \rangle (i \wedge \varphi) \\
&\vdash \langle \alpha \rangle (i \wedge \varphi) \rightarrow \langle \alpha \rangle (@_i \varphi) && (\text{@-intro'}) \\
&\vdash \langle \alpha \rangle (@_i \varphi) \rightarrow \@_i \varphi && (\text{back})
\end{aligned}$$

(*name'*). Suppose $\vdash i \rightarrow \varphi$. Then:

$$\begin{aligned}
&\vdash \@_i (i \rightarrow \varphi) && (\text{Nec}) \\
&\vdash \@_i i \rightarrow \@_i \varphi && (K) \\
&\vdash \@_i \varphi && (\text{@-refl.}) \\
&\vdash \varphi && (\text{name})
\end{aligned}$$

□

3.2. Extended Axiomatic Systems. In the next section we will prove that not only HXP is complete with respect to the class of all models, but that extensions of HXP with *pure axioms* and *existential saturation rules* preserve completeness with respect to the corresponding class of models. In this section we present such extensions.

The use of nominals, and in particular pure axioms and existential saturation rules, allows us to characterize classes of models that are not definable without them. For instance, the hybrid axiom $\@_i \neg \langle \mathbf{a} \rangle i$ forces the accessibility relation related to \mathbf{a} in the model to be *irreflexive*, which cannot be expressed in the basic modal logic. In this way, we can express properties about the underlying topology of a data model and also impose restrictions about the data fields. For instance, the axiom $\langle \@_i =_e \@_j \rangle \rightarrow \langle \@_i =_d \@_j \rangle$ for *data inclusion* that will be discussed in Section 3.5, expresses that if two nodes coincide in the data field e , then they must also coincide in the data field d .

Standard Translation. In order to establish frame conditions associated to pure axioms and existential saturation rules we define the *standard translation* of $\text{HXPath}_=$ into first-order logic, by mutual recursion between NExp and PExp .

Definition 3.3. The correspondence language for expressions of $\text{HXPath}_=$ is a relational language with a unary relation symbol P_i for each $p_i \in \text{Prop}$, a binary relation symbol $R_{\mathbf{a}}$ for each $\mathbf{a} \in \text{Mod}$ and a binary relation symbol D_e for each $e \in \text{Eq}$. Moreover, for each nominal $i \in \text{Nom}$ we will associate an indexed variable x_i . The function ST'_x from $\text{HXPath}_=$ -formulas

into its correspondence language is defined by

$$\begin{aligned}
\text{ST}'_x(i) &= x = x_i \\
\text{ST}'_x(p) &= P(x) \\
\text{ST}'_x(\neg\varphi) &= \neg\text{ST}'_x(\varphi) \\
\text{ST}'_x(\varphi \wedge \psi) &= \text{ST}'_x(\varphi) \wedge \text{ST}'_x(\psi) \\
\text{ST}'_x(\langle\alpha =_e \beta\rangle) &= \exists y, z. \text{ST}'_{x,y}(\alpha) \wedge \text{ST}'_{x,z}(\beta) \wedge D_e(y, z) \\
\text{ST}'_x(\langle\alpha \neq_e \beta\rangle) &= \exists y, z. \text{ST}'_{x,y}(\alpha) \wedge \text{ST}'_{x,z}(\beta) \wedge \neg D_e(y, z),
\end{aligned}$$

where y, z are first-order variables which have not been used yet in the translation, and where $\text{ST}'_{x,y}$ is defined as

$$\begin{aligned}
\text{ST}'_{x,y}(a) &= R_a(x, y) \\
\text{ST}'_{x,y}(@_i) &= y = x_i \\
\text{ST}'_{x,y}([\varphi]) &= x = y \wedge \text{ST}'_y(\varphi) \\
\text{ST}'_{x,y}(\alpha\beta) &= \exists z. \text{ST}'_{x,z}(\alpha) \wedge \text{ST}'_{z,y}(\beta),
\end{aligned}$$

with z not used yet in the translation. Finally, let $Eq(e)$ be the first-order formula stating that the relation D_e is an equivalence relation (i.e., reflexive, symmetric and transitive), we define the *standard translation* ST_x of a formula φ as

$$\text{ST}_x(\varphi) = \bigwedge \{Eq(e) \mid =_e \text{ appearing in } \varphi\} \wedge \text{ST}'_x(\varphi).$$

Since the standard translation mimics the semantic clauses for HXPath₌-formulas, the following proposition holds.

Proposition 3.4. *Let φ be an HXPath₌-formula, and let $\mathcal{M} = \langle M, \{\sim_e\}_{e \in \text{Eq}}, \{R_a\}_{a \in \text{Mod}}, V, \text{nom} \rangle$ be an abstract hybrid data model with $m \in M$. Then $\mathcal{M}, m \models \varphi$ if and only if $\mathcal{M}, g \models \text{ST}_x(\varphi)$, where g is an arbitrary first-order assignment such that $g(x_i) = \text{nom}(i)$ and $g(x) = m$.*

Notice, in the proposition above, that $\mathcal{M} = \langle M, \{\sim_e\}_{e \in \text{Eq}}, \{R_a\}_{a \in \text{Mod}}, V, \text{nom} \rangle$ is used to interpret $\text{ST}_x(\varphi)$ with $V(p)$ interpreting the unary relation symbol P , each accessibility relation R_a interpreting the binary relation symbol R_a and each relation \sim_e interpreting the binary relation D_e .

Pure Axioms and Existential Saturation Rules. We have now all the ingredients needed to define pure axioms and existential saturation rules, together with their associated frame conditions.

Definition 3.5 (Pure Axioms). We say that a formula (or in particular, an axiom) is *pure* if it does not contain any occurrences of propositional symbols. Let Π be a set of pure axioms, we define $\text{FC}(\Pi)$, the *frame condition associated to Π* as the universal closure of the standard translation of the axioms in Π , i.e.,

$$\text{FC}(\Pi) = \bigwedge \{ \forall x. \forall x_{i_1} \dots \forall x_{i_n}. \text{ST}_x(\varphi) \mid \varphi \in \Pi, i_1, \dots, i_n \text{ all the nominals in } \varphi \}.$$

Definition 3.6 (Existential Saturation Rules). Let $\varphi(i_1, \dots, i_n, j_1, \dots, j_m)$ be an HXPath₌ formula with no propositional symbols such that $i_1, \dots, i_n, j_1, \dots, j_m$ is an enumeration of all nominals appearing in φ . An *existential saturation rule* is a rule of the form

$$\frac{\vdash \varphi(i_1, \dots, i_n, j_1, \dots, j_m) \rightarrow \psi}{\vdash \psi} \quad \text{provided that } j_1, \dots, j_m \text{ do not occur in } \psi.$$

We will call φ the *head* of the rule, i_1, \dots, i_n its *universally quantified nominals* (notation: φ_{\forall}), and j_1, \dots, j_m its *existentially quantified nominals* (notation: φ_{\exists})¹. Let P be a set of existential saturation rules, we define $FC(P)$, the *frame condition associated to P* as follows:

$$FC(P) = \bigwedge \{ \forall x. \forall x_{i_1} \dots \forall x_{i_n}. \exists x_{j_1} \dots \exists x_{j_m}. ST_x(\varphi) \mid \varphi \text{ is the head of a rule } \rho \in P, \\ i_k \in \varphi_{\forall} \text{ and } j_l \in \varphi_{\exists} \}.$$

Let Π be a set of pure formulas and let P be a set of existential saturation rules, in what follows we write $HXP + \Pi + P$ for the axiomatic system HXP extended with Π as additional axioms and P as additional inference rules. In the coming sections we will show that this system is strongly complete with respect to the class of models based on frames satisfying the frame condition $FC(\Pi) \wedge FC(P)$.

Example 3.7. The axiom $p \rightarrow [a]\langle b \rangle p$ is not pure, since it contains the propositional symbol p . On the other hand, $\Pi = \{i \rightarrow [a]\langle b \rangle i, i \rightarrow [b]\langle a \rangle i\}$ is a set of pure axioms. Its corresponding frame condition $FC(\Pi)$ indicates that R_b is the inverse of R_a . Indeed, $FC(\Pi)$ is equivalent to $\forall x. \forall y. (R_a(x, y) \leftrightarrow R_b(y, x))$.

Consider the formula $@_i\langle a \rangle i$. It can be used to define two different existential saturation rules:

$$\rho_1 = \frac{\vdash @_i\langle a \rangle i \rightarrow \psi}{\vdash \psi} \quad (\text{no side condition})$$

and

$$\rho_2 = \frac{\vdash @_i\langle a \rangle i \rightarrow \psi}{\vdash \psi} \quad \text{where } i \text{ does not occur in } \psi.$$

$FC(\{\rho_1\})$ is equivalent to $\forall x. R_a(x, x)$ while $FC(\{\rho_2\})$ is equivalent to $\exists x. R_a(x, x)$.

Consider now the rule corresponding to the *Church-Rosser property* discussed in [BtC06]:

$$\frac{\vdash (@_i\langle a \rangle j \wedge @_i\langle a \rangle k \rightarrow @_j\langle a \rangle l \wedge @_k\langle a \rangle l) \rightarrow \psi}{\vdash \psi} \quad \text{where } l \text{ does not occur in } \psi.$$

The frame condition corresponding to this rule is:

$$\forall x. \forall x_i. \forall x_j. \forall x_k. \exists x_l. ST_x(@_i\langle a \rangle j \wedge @_i\langle a \rangle k \rightarrow @_j\langle a \rangle l \wedge @_k\langle a \rangle l),$$

which is equivalent to $\forall x_i. \forall x_j. \forall x_k. \exists x_l. (R_a(x_i, x_j) \wedge R_a(x_i, x_k) \rightarrow R_a(x_j, x_l) \wedge R_a(x_k, x_l))$.

It is not difficult to see that any pure axiom π is equivalent to the rule that uses π as head without side conditions (i.e., π_{\exists} is empty). In that sense, any axiomatic system $HXP + \Pi + P$ is equivalent to some $HXP + P'$, as pure axioms do not introduce additional expressive power. On the other hand, properties that mix both universal and existential quantification like the Church-Rosser property mentioned above cannot be captured using only pure axioms. However, axioms are simpler than rules with side conditions; and pure axioms are expressive enough to characterize many interesting properties.

¹Notice that given a particular φ , different existential saturation rules can be defined depending on which nominals are listed in its side condition.

3.3. Soundness and Completeness. It is a fairly straightforward exercise to prove that the axioms and rules of HXP are sound with respect to the class of all models. Similarly, any set Π of pure axioms is sound with respect to the class of models obtained from frames satisfying $FC(\Pi)$, and any set P of existential saturation rules is sound with respect to the class of frames satisfying $FC(P)$ (the proof is similar to the one provided in [BtC06]).

Theorem 3.8 (Soundness). *Let Π be a set of pure axioms and let P be a set of existential saturation rules. All the axioms and rules from HXP are valid over the class of abstract hybrid data models satisfying the frame condition $FC(\Pi) \wedge FC(P)$.*

Now we will devote ourselves to show that the axiomatic system is also strongly complete. The completeness argument follows the lines of the completeness proof for $\mathcal{HL}(@)$ and similar approaches (see, e.g., [Gol84, BtC06, SP10]), which is a Henkin-style proof with nominals playing the role of first-order constants.

We will prove in this section that the system $HXP + \Pi + P$ is *strongly complete* with respect to the class of abstract hybrid data models obtained from frames satisfying $FC(\Pi) \wedge FC(P)$. More precisely, given a particular extension $HXP + \Pi + P$, we will show that if $\mathfrak{C} = \text{Mod}(\mathfrak{F})$ for \mathfrak{F} the class of frames satisfying $FC(\Pi) \wedge FC(P)$, then $\Gamma \models_{\mathfrak{C}} \varphi$ implies $\Gamma \vdash_{HXP + \Pi + P} \varphi$, where $\Gamma \cup \{\varphi\}$ is a set of $HXPath_{=}$ -formulas. Or, equivalently, we need to show that every consistent set of formulas (for $HXP + \Pi + P$) is satisfiable in some abstract hybrid data model (in \mathfrak{C}). Recall that subscripts in relations \vdash and \models are omitted when they are clear from the context.

Definition 3.9. Let Γ be a set of formulas, we say that Γ is an $HXP + \Pi + P$ *maximal consistent set* (MCS for short) if and only if $\Gamma \not\vdash \perp$ and for all $\varphi \notin \Gamma$ we have $\Gamma \cup \{\varphi\} \vdash \perp$.

Proposition 3.10. *Let Γ be an MCS. Then, the following facts hold:*

- (1) *If $\{i, \varphi\} \subseteq \Gamma$ then $@_i \varphi \in \Gamma$.*
- (2) *If $@_i \langle \alpha =_e \beta \rangle \in \Gamma$ then $\langle @_i \alpha =_e @_i \beta \rangle \in \Gamma$.*
- (3) *If $\langle \alpha =_e @_i \beta \rangle \in \Gamma$ then $\langle \alpha =_e @_j @_i \beta \rangle \in \Gamma$.*

Proof. Item 1 is a consequence of $@\text{-intro}'$, 2 follows from $@\text{-dist}$ and 3 can be proved using *agree* and $=\text{-comm}$. \square

The next fact follows from the definition of MCS, as expected:

Fact 3.11. Let Γ be an MCS. Then for all φ , either $\varphi \in \Gamma$ or $\neg\varphi \in \Gamma$.

So far, we presented an axiom system together with the standard tools for proving its completeness. We also introduced non-orthodox rules (i.e., rules with side conditions), which will play a crucial role in the Henkin-style model we will build for proving completeness. The *paste* rule expresses that path expressions can control what happens in accessible states from a named state. The *name* rule says that if φ is provable to hold in an arbitrary state named by j , then φ is also provable. Now we introduce some properties that will be required in the construction of the Henkin-style model.

Definition 3.12 (Named, Pasted and ρ -saturated sets). Let Σ be a set of $HXPath_{=}$ -formulas.

- We say that Σ is *named* if for some nominal i we have that $i \in \Sigma$ (and we will say that Σ is named by i).
- We say that Σ is *pasted* if the following holds:
 - (1) $\langle @_i \mathbf{a} \alpha =_e \beta \rangle \in \Sigma$ implies that for some nominal j , $@_i \langle \mathbf{a} \rangle j \wedge \langle @_j \alpha =_e \beta \rangle \in \Sigma$.

- (2) $\langle @_i a \alpha \neq_e \beta \rangle \in \Sigma$ implies that for some nominal j , $\langle @_i \langle a \rangle j \wedge \langle @_j \alpha \neq_e \beta \rangle \rangle \in \Sigma$.
- For φ a formula and $i_1, \dots, i_n, j_1, \dots, j_n \in \text{Nom}$, let $\varphi[i_1/j_1, \dots, j_1/j_n]$ be the simultaneous substitution of i_k by j_k , $1 \leq k \leq n$ in φ . Let ρ be an existential saturation rule with head φ , $\varphi_{\forall} = i_1, \dots, i_n$ and $\varphi_{\exists} = j_1, \dots, j_m$. Then Σ is ρ -saturated if for all nominals $i'_1, \dots, i'_n \in \Sigma$, $\varphi[i_1/i'_1, \dots, i_n/i'_n, j_1/j'_1, \dots, j_m/j'_m] \in \Sigma$ for some $j'_1, \dots, j'_m \in \text{Nom}$. For P a set of existential saturation rules, Σ is P -saturated if it is ρ -saturated for all $\rho \in \text{P}$.

Now we are going to prove a crucial property in our completeness proof: the *Extended Lindenbaum Lemma*. Intuitively, it says that the rules of $\text{HXP} + \Pi + \text{P}$ allow us to extend MCSs to *named and pasted* MCSs, provided we enrich the language with new nominals. This lemma will be useful to obtain the models we need from an MCS.

Lemma 3.13 (Extended Lindenbaum Lemma). *Let Nom' be a (countably) infinite set of nominals disjoint from Nom , and let $\text{HXPath}'_{=}$ be the language obtained by adding these new nominals to $\text{HXPath}_{=}$. Then, every consistent set of formulas in $\text{HXPath}_{=}$ can be extended to a named and pasted MCS in $\text{HXPath}'_{=}$.*

Proof. Enumerate Nom' and let k be the first nominal in the enumeration. Given Σ a consistent set in $\text{HXPath}_{=}$, $\Sigma \cup \{k\}$ is consistent, otherwise for some conjunction θ from Σ , $\vdash k \rightarrow \neg\theta$. By the *name'* rule, $\vdash \neg\theta$, contradicting the consistency of Σ .

Now enumerate all formulas in $\text{HXPath}'_{=}$. Define $\Sigma^0 = \Sigma \cup \{k\}$ and suppose we have defined Σ^n , for $n \geq 0$. Let φ_{n+1} be the $(n+1)^{\text{th}}$ formula in the enumeration of $\text{HXPath}'_{=}$. Define Σ_{n+1} as follows. If $\Sigma^n \cup \{\varphi_{n+1}\}$ is inconsistent, then $\Sigma^{n+1} = \Sigma^n$. Otherwise:

- (1) $\Sigma^{n+1} = \Sigma^n \cup \{\varphi_{n+1}\}$ if φ_{n+1} is not of the form $\langle @_i a \alpha * \beta \rangle$.
- (2) $\Sigma^{n+1} = \Sigma^n \cup \{\varphi_{n+1}\} \cup \{ @_i \langle a \rangle j \wedge \langle @_j \alpha * \beta \rangle \}$, if φ_{n+1} is of the form $\langle @_i a \alpha * \beta \rangle$. Here j is the first nominal in the enumeration that does not occur in any formula of Σ^n or $\langle @_i a \alpha * \beta \rangle$.

Let $\Sigma^\omega = \bigcup_{n \geq 0} \Sigma^n$. This set is named (by k), maximal and pasted. Furthermore, it is consistent as a direct consequence of the *paste* rule. \square

Lemma 3.14 (Rule Saturation Lemma). *Let Nom' be a (countably) infinite set of nominals disjoint from Nom , and let $\text{HXPath}'_{=}$ be the language obtained by adding these new nominals to $\text{HXPath}_{=}$. Let Π be a set of pure axioms, and P be a set of existential saturation rules. Then, every consistent set of formulas in $\text{HXPath}_{=}$ can be extended to a named, pasted and P -saturated MCS in $\text{HXPath}'_{=}$.*

Proof. Let Σ be a consistent set of formulas in $\text{HXPath}_{=}$. First, we will show that Σ can be extended to some Σ^+ such that for every rule $\rho \in \text{P}$ with head φ , $\varphi_{\forall} = i_1 \dots i_n$, and $\varphi_{\exists} = j_1 \dots j_m$, and for all $i'_1, \dots, i'_n \in \Sigma$, there are nominals j'_1, \dots, j'_m such that $\varphi(i_1/i'_1, \dots, i_n/i'_n, j_1/j'_1, \dots, j_m/j'_m) \in \Sigma^+$.

Let Nom' be a (countably) infinite set of nominals disjoint from Nom . Given Σ and P , say that a pair (ρ, \bar{i}) is well-formed if $\rho \in \text{P}$ with head φ , and \bar{i} is a sequence of nominals in Σ with length equal to the length of φ_{\forall} . For Σ and P countable, the set of well-formed pairs is countable and can be enumerated. Define $\Sigma^0 = \Sigma$. Let $(\rho_{n+1}, i'_1 \dots i'_k)$ be the $(n+1)^{\text{th}}$ pair in the enumeration. Define Σ^{n+1} as follows.

- $$\Sigma^{n+1} = \Sigma^n \cup \{ \varphi(i_1/i'_1, \dots, i_k/i'_k, j_1/j'_1, \dots, j_m/j'_m) \}$$
- where φ is the head of ρ_{n+1} , $\varphi_{\forall} = i_1, \dots, i_k$, $\varphi_{\exists} = j_1, \dots, j_m$, and j'_1, \dots, j'_m are the first m nominals in Nom' that do not occur in any formula of Σ^n .

We define $\Sigma^+ = \bigcup_{n \geq 0} \Sigma^n$, which is consistent and extends Σ as described above.

Now consider any consistent set of formulas Γ . Let $\Gamma^0 = \Gamma$, and for all $n \geq 0$ let Γ^{n+1} be a named and pasted MCS extending $(\Gamma^n)^+$ (which exists by Lemma 3.13). Then we have the following chain of inclusions:

$$\Gamma = \Gamma^0 \subseteq (\Gamma^0)^+ \subseteq \Gamma^1 \subseteq (\Gamma^1)^+ \subseteq \dots$$

Let $\Gamma^\omega = \bigcup_{n \geq 0} \Gamma^n$. Then Γ^ω is a named, pasted and P-saturated MCS in HXPath' . As we used only countably many new nominals, this set is also countable. \square

From a named and pasted MCS we can extract a model:

Definition 3.15 (Extracted Model). Let Γ be a named and pasted MCS². Define $\mathcal{M}_\Gamma = \langle M, \{\sim_e\}_{e \in \text{Eq}}, \{R_a\}_{a \in \text{Mod}}, V, \text{nom} \rangle$, the *extracted model* from Γ , as

- $M = \{\Delta_i \mid \Delta_i = \{j \mid @_i j \in \Gamma\}\}$
- $\Delta_i R_a \Delta_j$ iff $@_i \langle a \rangle j \in \Gamma$
- $p \in V(\Delta_i)$ iff $@_i p \in \Gamma$
- $\text{nom}(i) = \Delta_i$
- $\Delta_i \sim_e \Delta_j$ iff $@_i \langle \epsilon =_e @_j \rangle \in \Gamma$.

We need to prove that \mathcal{M}_Γ is well defined, and that it is actually an abstract hybrid data model.

Proposition 3.16.

- (1) $\Delta_i = \Delta_j$ implies $@_i \varphi \in \Gamma$ iff $@_j \varphi \in \Gamma$.
- (2) For all $e \in \text{Eq}$, \sim_e is an equivalence relation.

Proof. Item 1 ensures that the definition of \mathcal{M}_Γ does not depend of the particular nominal taken as representative of Δ_i . The property follows directly from *bridge*.

For item 2, we prove:

- *Reflexivity*: We need to show that $\Delta_i \sim_e \Delta_i$, which by definition is equivalent to $@_i \langle \epsilon =_e @_i \rangle \in \Gamma$. By *equal* we have $\langle \epsilon =_e \epsilon \rangle \in \Gamma$, and applying *Nec* we get $@_i \langle \epsilon =_e \epsilon \rangle \in \Gamma$. Also, by *comp=-dist*, $\langle @_i = @_i \rangle \in \Gamma$, and by *agree* $\langle @_i =_e @_i @_i \rangle \in \Gamma$. Then, by *@=-dist*, $@_i \langle \epsilon =_e @_i \rangle \in \Gamma$, as wanted.

- *Symmetry*: $\Delta_i \sim_e \Delta_j$ iff $@_i \langle \epsilon =_e @_j \rangle \in \Gamma$. By *neutral* and *=-comm* we get $@_i \langle \epsilon =_e @_j \epsilon \rangle \in \Gamma$. Then, by *@-swap* $@_j \langle \epsilon =_e @_i \epsilon \rangle \in \Gamma$. Therefore (by *neutral*) $@_j \langle \epsilon =_e @_i \rangle \in \Gamma$.

- *Transitivity*: Suppose $\Delta_i \sim_e \Delta_j$ and $\Delta_j \sim_e \Delta_k$, iff $@_i \langle \epsilon =_e @_j \rangle \in \Gamma$ and $@_j \langle \epsilon =_e @_k \rangle \in \Gamma$. This means that we have (by *@-swap*) $@_j \langle \epsilon =_e @_i \rangle \in \Gamma$, and $@_j \langle \epsilon =_e @_k \rangle \in \Gamma$. Then $@_j (\langle \epsilon =_e @_i \rangle \wedge \langle \epsilon =_e @_k \rangle) \in \Gamma$ (let us call this fact (\dagger) for later use). On the other hand, $\langle \epsilon =_e @_i \rangle \wedge \langle \epsilon =_e @_k \rangle \rightarrow \langle @_i =_e @_k \rangle \in \Gamma$ because of ϵ -*trans*. Then (by *Nec*) $@_j (\langle \epsilon =_e @_i \rangle \wedge \langle \epsilon =_e @_k \rangle) \rightarrow \langle @_i =_e @_k \rangle \in \Gamma$, and by *K* and distributivity of $@$ with respect to \wedge , $@_j \langle \epsilon =_e @_i \rangle \wedge @_j \langle \epsilon =_e @_k \rangle \rightarrow @_j \langle @_i =_e @_k \rangle \in \Gamma$. By Modus Ponens with (\dagger) we have $@_j \langle @_i =_e @_k \rangle \in \Gamma$, and by *comp=-dist* $\langle @_j @_i =_e @_j @_k \rangle \in \Gamma$. Using *back* and *agree* we get $\langle @_i =_e @_i @_k \rangle \in \Gamma$. Hence by *@=-dist*, $@_i \langle \epsilon =_e @_k \rangle \in \Gamma$, which gives us $\Delta_i \sim_e \Delta_k$. \square

Proposition 3.17. Let $\mathcal{M}_\Gamma = \langle M, \{\sim_e\}_{e \in \text{Eq}}, \{R_a\}_{a \in \text{Mod}}, V, \text{nom} \rangle$ be the extracted model, for some Γ . Then,

- (1) $\Delta_i \not\sim_e \Delta_j$ if and only if $@_i \langle \epsilon \neq_e @_j \rangle \in \Gamma$,
- (2) if $\Delta_i R_a \Delta_j$ then for all $@_j \varphi \in \Gamma$, $@_i \langle a \rangle \varphi \in \Gamma$.

²For this definition, it is irrelevant whether Γ is P-saturated.

Proof. Item 1 follows from @-data ; for item 2 suppose $\Delta_i R_a \Delta_j$, then we have $\text{@}_i \langle \mathbf{a} \rangle j \in \Delta_i$. Let $\text{@}_j \varphi \in \Gamma$, then by *comp-dist* we have $\langle \text{@}_i \mathbf{a} \rangle j \in \Gamma$, hence by *bridge* we get $\langle \text{@}_i \mathbf{a} \rangle \varphi \in \Gamma$. Therefore, by @*-dist , $\text{@}_i \langle \mathbf{a} \rangle \varphi \in \Gamma$. \square

Now, given a named and pasted MCS Γ we can prove the following lemma:

Lemma 3.18 (Existence Lemma). *Let Γ be an MCS and $\mathcal{M}_\Gamma = \langle M, \{\sim_e\}_{e \in \text{Eq}}, \{R_a\}_{a \in \text{Mod}}, V, \text{nom} \rangle$ be the extracted model from Γ . Let $\Delta_i \in M$, then*

- (1) $\text{@}_i \langle \mathbf{a} \alpha =_e \beta \rangle \in \Gamma$ implies there exists $\Delta_j \in M$ s.t. $\Delta_i R_a \Delta_j$ and $\text{@}_j \langle \alpha =_e \text{@}_i \beta \rangle \in \Gamma$.
- (2) $\text{@}_i \langle \mathbf{a} \alpha \neq_e \beta \rangle \in \Gamma$ implies there exists $\Delta_j \in M$ s.t. $\Delta_i R_a \Delta_j$ and $\text{@}_j \langle \alpha \neq_e \text{@}_i \beta \rangle \in \Gamma$.
- (3) $\text{@}_i \langle \text{@}_j \alpha =_e \text{@}_k \beta \rangle \in \Gamma$ implies $\text{@}_j \langle \alpha =_e \text{@}_k \beta \rangle \in \Gamma$.
- (4) $\text{@}_i \langle \text{@}_j \alpha \neq_e \text{@}_k \beta \rangle \in \Gamma$ implies $\text{@}_j \langle \alpha \neq_e \text{@}_k \beta \rangle \in \Gamma$.

Proof. First, we discuss 1 (2 is similar). We suppose $\text{@}_i \langle \mathbf{a} \alpha =_e \beta \rangle \in \Gamma$. Then, by @=-dist , $\langle \text{@}_i \mathbf{a} \alpha =_e \text{@}_i \beta \rangle \in \Gamma$. Because Γ is pasted, $\text{@}_i \langle \mathbf{a} \rangle j \wedge \langle \text{@}_j \alpha =_e \text{@}_i \beta \rangle \in \Gamma$. As Γ is an MCS, $\text{@}_i \langle \mathbf{a} \rangle j \in \Gamma$ and $\langle \text{@}_j \alpha =_e \text{@}_i \beta \rangle \in \Gamma$. By definition of \mathcal{M}_Γ we obtain $\Delta_i R_a \Delta_j$, and by *agree*, we have $\langle \text{@}_j \alpha =_e \text{@}_j \text{@}_i \beta \rangle \in \Gamma$. Then, $\text{@}_j \langle \alpha =_e \text{@}_i \beta \rangle \in \Gamma$ by @=-dist .

For 3 (4 is similar) assume $\text{@}_i \langle \text{@}_j \alpha =_e \text{@}_k \beta \rangle \in \Gamma$, then by *comp=-dist*, $\langle \text{@}_i \text{@}_j \alpha =_e \text{@}_i \text{@}_k \beta \rangle \in \Gamma$. Applying *agree* twice, we have $\langle \text{@}_j \alpha =_e \text{@}_j \text{@}_k \beta \rangle \in \Gamma$, then by @=-dist $\text{@}_j \langle \alpha =_e \text{@}_k \beta \rangle \in \Gamma$. \square

Corollary 3.19. *Let Γ be an MCS and let $\mathcal{M}_\Gamma = \langle M, \{\sim_e\}_{e \in \text{Eq}}, \{R_a\}_{a \in \text{Mod}}, V, \text{nom} \rangle$ be the extracted model, and $\Delta_i \in M$. If $\text{@}_i \langle \mathbf{a} \alpha \rangle \varphi \in \Gamma$, then there exists $\Delta_j \in M$ such that $\Delta_i R_a \Delta_j$ and $\text{@}_j \langle \alpha \rangle \varphi \in \Gamma$.*

Proof. Let $\Delta_i \in M$. By hypothesis, $\text{@}_i \langle \mathbf{a} \alpha [\varphi] =_e \mathbf{a} \alpha [\varphi] \rangle \in \Gamma$, then by Lemma 3.18 there exists $\Delta_j \in M$ such that $\Delta_i R_a \Delta_j$ and $\text{@}_j \langle \alpha [\varphi] =_e \text{@}_i \alpha [\varphi] \rangle \in \Gamma$. By *comp=-dist*, $\langle \text{@}_j \alpha [\varphi] =_e \text{@}_j \text{@}_i \alpha [\varphi] \rangle \in \Gamma$, and by *comp-neutral* and *subpath* we get $\langle \text{@}_j \alpha [\varphi] \rangle \top \in \Gamma$. Then, using *comp-dist*, *comp-assoc* and *=-test*, we have $\text{@}_j \langle \alpha \rangle \varphi \in \Gamma$. \square

Now we are ready to prove the Truth Lemma that states that membership in an MCS generating an extracted model is equivalent to being true at a state in the extracted model. First let us introduce a notion of size for node and path expressions, which we will use in the inductive cases of the proof.

Definition 3.20. We define inductively the size of a path and node expression (notation $|\cdot|$) as follows:

$$\begin{array}{ll}
|\mathbf{a}| &= 2 & |p| &= 1, p \in \text{Prop} \cup \text{Nom} \\
|\text{@}_i| &= 1 & |\neg \varphi| &= |\varphi| \\
|[\varphi]| &= 1 + |\varphi| & |\varphi \wedge \psi| &= |\varphi| + |\psi| \\
|\alpha \beta| &= |\alpha| + |\beta| & |\langle \alpha * \beta \rangle| &= |\alpha| + |\beta|,
\end{array}$$

where α, β are path expressions and φ, ψ are node expressions.

Lemma 3.21 (Truth Lemma). *Let $\mathcal{M}_\Gamma = \langle M, \{\sim_e\}_{e \in \text{Eq}}, \{R_a\}_{a \in \text{Mod}}, V, \text{nom} \rangle$ be the extracted model from an MCS Γ , and let $\Delta_i \in M$. Then, for any formula φ ,*

$$\mathcal{M}_\Gamma, \Delta_i \models \varphi \text{ iff } \text{@}_i \varphi \in \Gamma.$$

Proof. In fact we will prove a stronger result. Let $\Delta_i, \Delta_j \in M$, φ be a node expression and α be a path expression.

(IH1): $\mathcal{M}_\Gamma, \Delta_i \models \varphi$ iff $\text{@}_i \varphi \in \Gamma$.

(IH2): $\mathcal{M}_\Gamma, \Delta_i, \Delta_j \models \alpha$ iff $@_i\langle\alpha\rangle j \in \Gamma$.

The proof is a double induction argument, proceeding first on the structural complexity of φ and α , and then on the size of path-formulas as per Definition 3.20. First, we prove the base cases:

- $\alpha = \mathbf{a}$: Suppose $\mathcal{M}_\Gamma, \Delta_i, \Delta_j \models \mathbf{a}$, iff $\Delta_i R_a \Delta_j$ (by \models), iff $@_i\langle\mathbf{a}\rangle j \in \Gamma$ (by definition of extracted model).
- $\alpha = @_k$: Suppose $\mathcal{M}_\Gamma, \Delta_i, \Delta_j \models @_k$, iff $nom(k) = \Delta_j$. But by definition of nom , $\Delta_j = \Delta_k$, and because we know that $j \in \Delta_j$ we have $j \in \Delta_k$. Then, we have $@_k j \in \Gamma$, and by *agree*, $@_i @_k j \in \Gamma$.
- $\varphi = p$: $\mathcal{M}_\Gamma, \Delta_i \models p$ iff $p \in V(\Delta_i)$, iff $@_i p \in \Gamma$.
- $\varphi = j$: $\mathcal{M}_\Gamma, \Delta_i \models j$ iff $nom(j) = \Delta_i$, iff $\Delta_i = \Delta_j$ iff $j \in \Delta_i$, iff $@_i j \in \Gamma$.

Now we prove the inductive cases:

- $\varphi = \psi \wedge \theta$ and $\varphi = \neg\psi$ are direct from (IH1).
- $\alpha = [\psi]$: $\mathcal{M}_\Gamma, \Delta_i, \Delta_j \models [\psi]$ iff $\Delta_i = \Delta_j$ and $\mathcal{M}_\Gamma, \Delta_i \models \psi$. By (IH1), we have $@_i \psi \in \Gamma$ and $j \in \Delta_i$, then $@_i j \in \Gamma$. As Γ is an MCS, we have $@_i(\psi \wedge j) \in \Gamma$, and by idempotence of the conjunction we have $@_i(\psi \wedge \psi \wedge j \wedge j) \in \Gamma$. Also, $\langle \epsilon =_e \epsilon \rangle$ is a theorem, by *Nec* we have $@_i \langle \epsilon =_e \epsilon \rangle \in \Gamma$, then $@_i(\psi \wedge \psi \wedge j \wedge j \wedge \langle \epsilon =_e \epsilon \rangle) \in \Gamma$. Using *=-test* and *=-comm* we obtain $@_i\langle[\psi][j] =_e [\psi][j]\rangle \in \Gamma$ (which is the same as $@_i\langle[\psi]\rangle j$) as we wanted.
- $\alpha = \mathbf{a}\beta$: For the left to right implication, let $\mathcal{M}_\Gamma, \Delta_i, \Delta_j \models \mathbf{a}\beta$ iff there is some $\Delta_k \in M$ such that $\mathcal{M}_\Gamma, \Delta_i, \Delta_k \models \mathbf{a}$ and $\mathcal{M}_\Gamma, \Delta_k, \Delta_j \models \beta$. By (IH2), we have $@_i\langle\mathbf{a}\rangle k \in \Gamma$ and $@_k\langle\beta\rangle j \in \Gamma$, then $@_i\langle\mathbf{a}\rangle k \wedge @_k\langle\beta\rangle j \in \Gamma$. By *agree*, we have $@_i\langle\mathbf{a}\rangle k \wedge @_i @_k\langle\beta\rangle j \in \Gamma$, and with a simple hybrid logic argument, we get $@_i\langle(\mathbf{a})k \wedge @_k\langle\beta\rangle j\rangle \in \Gamma$. By *bridge*, we have $@_i\langle\mathbf{a}\rangle\langle\beta\rangle j \in \Gamma$, hence by *comp-dist* $@_i\langle\mathbf{a}\beta\rangle j \in \Gamma$.

For the other direction, assume $@_i\langle\mathbf{a}\beta\rangle j \in \Gamma$. Then, by Corollary 3.19 there exists some k such that $\Delta_i R_a \Delta_k$ and $@_k\langle\beta\rangle j \in \Gamma$. Then, applying (IH2) twice, we get $\mathcal{M}_\Gamma, \Delta_i, \Delta_k \models \mathbf{a}$ and $\mathcal{M}_\Gamma, \Delta_k, \Delta_j \models \beta$. Therefore, $\mathcal{M}_\Gamma, \Delta_i, \Delta_j \models \mathbf{a}\beta$ as needed.

- $\alpha = @_k\beta$ and $\alpha = [\psi]\beta$: these cases are similar to the previous one.

For node expressions of the form $\langle\alpha * \beta\rangle$ we need to do induction on the size of α and β . Notice that by **-comm*, $@_i\langle\alpha * \beta\rangle \in \Gamma$ iff $@_i\langle\beta * \alpha\rangle \in \Gamma$. And by the semantic definition, $\mathcal{M}_\Gamma, \Delta_i \models \langle\alpha * \beta\rangle$ iff $\mathcal{M}_\Gamma, \Delta_i \models \langle\beta * \alpha\rangle$. So we need only carry out the inductive steps for α . Moreover, by *comp-neutral*, $\vdash \langle\alpha * \beta\rangle \leftrightarrow \langle\alpha \epsilon * \beta\rangle$ which is also a validity. So we can assume that every path ends in a test. The base case then is when $|\alpha| + |\beta| = 2$, and both α and β are tests.

- $\varphi = \langle[\psi] =_e [\theta]\rangle$: direct from *test-dist*.
- $\varphi = \langle[\psi] \neq_e [\theta]\rangle$: By *test- \perp* the formula is equivalent to \perp , hence not in Γ . Moreover, by the defined semantics the formula is unsatisfiable.

Now, let us consider $|\alpha| + |\beta| \geq 3$:

- $\varphi = \langle\mathbf{a}\beta =_e \gamma\rangle$: Let us prove the right to left direction. Suppose $@_i\langle\mathbf{a}\beta =_e \gamma\rangle \in \Gamma$. By the Existence Lemma, there is $\Delta_j \in M$ such that $\Delta_i R_a \Delta_j$, and $@_j\langle\beta =_e @_i\gamma\rangle \in \Gamma$, for some $j \in \text{Nom}$. Notice that $|\@_j\langle\beta =_e @_i\gamma\rangle| \leq |\@_i\langle\mathbf{a}\beta =_e \gamma\rangle|$. Applying (IH1) we obtain $\mathcal{M}_\Gamma, \Delta_j \models \langle\beta =_e @_i\gamma\rangle$, so there exists $\Delta_1, \Delta_2 \in M$ such that

- (1) $\mathcal{M}_\Gamma, \Delta_j, \Delta_1 \models \beta$,
- (2) $\mathcal{M}_\Gamma, \Delta_j, \Delta_2 \models @_i\gamma$,

(3) $\Delta_1 \sim_e \Delta_2$.

From 1 and $\Delta_i R_a \Delta_j$ we get $\mathcal{M}_\Gamma, \Delta_i, \Delta_1 \models \mathbf{a}\beta$ and from 2 and the semantic interpretation of $@$ we get $\mathcal{M}_\Gamma, \Delta_i, \Delta_2 \models \gamma$. Then, together with 3 we get $\mathcal{M}_\Gamma, \Delta_i \models \langle \mathbf{a}\beta =_e \gamma \rangle$, as wanted.

For the other direction, suppose $\mathcal{M}_\Gamma, \Delta_i \models \langle \mathbf{a}\beta =_e \gamma \rangle$, iff there are Δ_j, Δ_k such that $\mathcal{M}_\Gamma, \Delta_i, \Delta_j \models \mathbf{a}\beta$, $\mathcal{M}_\Gamma, \Delta_i, \Delta_k \models \gamma$ and $\Delta_j \sim_e \Delta_k$. First notice that $\mathcal{M}_\Gamma, \Delta_i, \Delta_j \models \mathbf{a}\beta$ iff there exists $\Delta_t \in M$ s.t. $\mathcal{M}_\Gamma, \Delta_i, \Delta_t \models \mathbf{a}$ and $\mathcal{M}_\Gamma, \Delta_t, \Delta_j \models \beta$. Then we have, $\Delta_i R_a \Delta_t$, and by definition of \mathcal{M}_Γ we get $@_i \langle \mathbf{a} \rangle t \in \Gamma$. On the other hand, by (IH2) on $\mathcal{M}_\Gamma, \Delta_t, \Delta_j \models \beta$ we obtain $@_t \langle \beta \rangle j \in \Gamma$. Since $\{ @_i \langle \mathbf{a} \rangle t, @_t \langle \beta \rangle j \} \subseteq \Gamma$, by *comp-dist* we have:

(1) $@_i \langle \mathbf{a}\beta \rangle j \in \Gamma$.

By (IH2) on $\mathcal{M}_\Gamma, \Delta_i, \Delta_k \models \gamma$, and by definition of \mathcal{M}_Γ together with $\Delta_j \sim_e \Delta_k$ we obtain, respectively:

(2) $@_i \langle \gamma \rangle k \in \Gamma$, and

(3) $@_j \langle \epsilon =_e @_k \rangle \in \Gamma$.

By 1 and Corollary 3.19 there exists Δ_l such that

(4) $@_l \langle \beta \rangle j \in \Gamma$.

(\dagger) From 2 we have $@_i \langle \gamma \rangle k \in \Gamma$ and from 3 we can obtain $@_k \langle \epsilon =_e @_j \rangle \in \Gamma$, then we have $\langle @_i \gamma \rangle k \wedge @_k \langle \epsilon =_e @_j \rangle \in \Gamma$, by *comp-dist*. By *bridge*, $\langle @_i \gamma \rangle \langle \epsilon =_e @_j \rangle \in \Gamma$, then by *comp=-dist* and *back*, we get $\langle @_i \gamma =_e @_j \rangle \in \Gamma$. Applying *=-comm*, *comp-neutral*, *agree* and *@-dist*, $@_j \langle \epsilon =_e @_i \gamma \rangle \in \Gamma$.

Also, from 4 we have $@_l \langle \beta \rangle j \in \Gamma$, then $@_j \langle \epsilon =_e @_i \gamma \rangle \wedge \langle @_l \beta \rangle j \in \Gamma$ (by MCS and *comp-dist*), and by *bridge* we get $\langle @_l \beta \rangle \langle \epsilon =_e @_i \gamma \rangle \in \Gamma$. By *comp=-dist* and *comp-neutral*, $\langle @_l \beta =_e @_l \beta @_i \gamma \rangle \in \Gamma$, then by *back*, *agree* and *@=-dist* we have $@_l \langle \beta =_e @_i \gamma \rangle \in \Gamma$. Then we have

$$\begin{aligned}
& @_l \langle \beta =_e @_i \gamma \rangle \in \Gamma \\
\Rightarrow & @_i \langle \mathbf{a} \rangle \langle \beta =_e @_i \gamma \rangle \in \Gamma && (\Delta_i R_a \Delta_l, \text{Prop. 3.17 item 2}) \\
\Rightarrow & @_i \langle \mathbf{a}\beta =_e \mathbf{a}@_i \gamma \rangle \in \Gamma && (\text{comp}=-\text{dist}) \\
\Rightarrow & @_i \langle \mathbf{a}\beta =_e @_i \gamma \rangle \in \Gamma && (\text{back}) \\
\Rightarrow & \langle @_i \mathbf{a}\beta =_e @_i @_i \gamma \rangle \in \Gamma && (\text{comp}=-\text{dist}) \\
\Rightarrow & \langle @_i \mathbf{a}\beta =_e @_i \gamma \rangle \in \Gamma && (\text{back}) \\
\Rightarrow & @_i \langle \mathbf{a}\beta =_e \gamma \rangle \in \Gamma && (@=-\text{dist})
\end{aligned}$$

- $\varphi = \langle [\psi]\beta =_e \gamma \rangle$: Let us prove the right to left direction. Suppose $@_i \langle [\psi]\beta =_e \gamma \rangle \in \Gamma$. By *=-test*, we have $@_i (\psi \wedge \langle \beta =_e \gamma \rangle) \in \Gamma$, iff (by using an argument purely based on modal reasoning), $@_i \psi \in \Gamma$ and $@_i \langle \beta =_e \gamma \rangle \in \Gamma$. By (IH1) on both expressions we get $\mathcal{M}_\Gamma, \Delta_i \models \psi$ and $\mathcal{M}_\Gamma, \Delta_i \models \langle \beta =_e \gamma \rangle$. Then, by the semantic interpretation of \wedge and $[\psi]$, we get $\mathcal{M}_\Gamma, \Delta_i \models \langle [\psi]\beta =_e \gamma \rangle$, as wanted.

For the other direction, suppose $\mathcal{M}_\Gamma, \Delta_i \models \langle [\psi]\beta =_e \gamma \rangle$, iff there are Δ_j, Δ_k such that $\mathcal{M}_\Gamma, \Delta_i, \Delta_j \models [\psi]\beta$, $\mathcal{M}_\Gamma, \Delta_i, \Delta_k \models \gamma$ and $\Delta_j \sim_e \Delta_k$. Then, by (IH2) and definition of \mathcal{M}_Γ we have:

(1) $@_i \langle \beta \rangle j \in \Gamma$,

(2) $@_i \langle \gamma \rangle k \in \Gamma$,

(3) $@_j \langle \epsilon =_e @_k \rangle \in \Gamma$, and

(4) $@_i \psi \in \Gamma$.

Using the same argument as in (‡) the proof that $@_i\langle[\psi]\beta =_e \gamma\rangle \in \Gamma$ is straightforward.
- $\varphi = \langle @_j\beta =_e \gamma \rangle$: For the left to right direction suppose $\mathcal{M}_\Gamma, \Delta_i \models \langle @_j\beta =_e \gamma \rangle$, iff there are Δ_k, Δ_l such that $\mathcal{M}_\Gamma, \Delta_i, \Delta_k \models @_j\beta$, $\mathcal{M}_\Gamma, \Delta_i, \Delta_l \models \gamma$ and $\Delta_k \sim_e \Delta_l$. Then, by (IH2) and definition of \mathcal{M}_Γ we have:

- (1) $@_i\langle @_j\beta \rangle k \in \Gamma$ iff $@_j\langle \beta \rangle k \in \Gamma$,
- (2) $@_i\langle \gamma \rangle l \in \Gamma$, and
- (3) $@_l\langle \epsilon =_e @_k \rangle \in \Gamma$, iff $@_k\langle \epsilon =_e @_l \rangle \in \Gamma$.

By 1 and 3, applying *bridge* we get $@_j\langle \beta \rangle \langle \epsilon =_e @_l \rangle \in \Gamma$, iff (by *comp=-dist*) $@_j\langle \beta =_e \beta @_l \rangle \in \Gamma$. By *back*, we get $@_j\langle \beta =_e @_l \rangle \in \Gamma$, which is equivalent to $@_l\langle \epsilon =_e @_j\beta \rangle \in \Gamma$ (by *agree* and *comp=-dist*). Together with 2 and *bridge* we get $@_i\langle \gamma \rangle \langle \epsilon =_e @_j\beta \rangle \in \Gamma$, hence $@_i\langle \gamma =_e \gamma @_j\beta \rangle \in \Gamma$ iff (by *back* and *=-comm*) $@_i\langle @_j\beta =_e \gamma \rangle \in \Gamma$.

For the other direction suppose $@_i\langle @_j\beta =_e \gamma \rangle \in \Gamma$. First notice that, in all the cases we considered so far, the induction is on the path expression appearing on the left side of the $=$. However, an analogous argument can be applied if we do induction in the path expression on the right side of the $=$, by *=-comm*. Suppose we proceed as above for the node expression $\langle @_j\beta =_e \gamma \rangle$. If we apply the exact same steps, we will find out that this time we need to do induction on γ ; but, as we mentioned, the cases for **a** and $[\varphi]$ are symmetric in both sides of the $=$. As a consequence, it all boils down to consider only the case $\gamma = @_k\eta$.

Suppose $@_i\langle @_j\beta =_e @_k\eta \rangle \in \Gamma$. By Existence Lemma we have $@_j\langle \beta =_e @_k\eta \rangle \in \Gamma$, hence by (IH1) $\mathcal{M}_\Gamma, \Delta_j \models \langle \beta =_e @_k\eta \rangle$. By semantics of $@$, and the fact that \mathcal{M}_Γ is named, $\mathcal{M}_\Gamma, \Delta_i \models \langle @_j\beta =_e @_k\eta \rangle$.

- The cases involving \neq_e are analogous, using item 1 from Proposition 3.17 to obtain $@_j\langle \epsilon =_e @_k \rangle \notin \Gamma$ in item 3 above. \square

Lemma 3.22 (Frame Lemma). *If Γ is a named, pasted and P-saturated MCS of HXP+ Π +P, then the underlying frame of \mathcal{M}_Γ satisfies $FC(\Pi) \wedge FC(P)$.*

Proof. Since \mathcal{M}_Γ is a named model and Γ contains all instances of elements of Π , it follows that the underlying frame of \mathcal{M}_Γ satisfies $FC(\Pi)$. Since \mathcal{M}_Γ is a named model and Γ is P-saturated, it follows that the underlying frame of \mathcal{M}_Γ satisfies $FC(P)$. \square

As a result we obtain the completeness result.

Theorem 3.23 (Strong Completeness). *Let Π be a set of pure axioms and P a set of existential saturation rules. Let $\mathfrak{C} = \text{Mod}(\mathfrak{F})$ for \mathfrak{F} the class of all frames satisfying $FC(\Pi) \wedge FC(P)$. Then, the axiomatic system HXP + Π + P is strongly complete for \mathfrak{C} .*

Proof. We need to prove that every set of HXP $_{\text{path}=\text{-}}$ -formulas Σ is consistent if and only if Σ is satisfiable in an abstract hybrid data model satisfying the frame properties defined by Π and P.

For any consistent Σ , we can use the Rule Saturation Lemma to obtain Σ^ω , which is a named, pasted and P-saturated MCS in HXP $_{\text{path}=\text{-}}$ extended by a set Nom' of additional nominals. Let $\mathcal{M}_{\Sigma^\omega} = \langle M, \{\sim_e\}_{e \in \text{Eq}}, \{R_a\}_{a \in \text{Mod}}, V, \text{nom} \rangle$ be the extracted model from Σ^ω . Let $i \in \Sigma^\omega$, for all $\varphi \in \Sigma$, by *@-intro* we have $@_i\varphi \in \Sigma^\omega$ since Σ^ω is MCS. Then by the Truth Lemma, $\mathcal{M}_{\Sigma^\omega}, \Delta_i \models \varphi$. By the Frame Lemma, $\mathcal{M}_{\Sigma^\omega}$ satisfies all required frame properties. \square

Because the class of abstract data models is a conservative abstraction of concrete data models, we can conclude:

Corollary 3.24. *The axiomatic system $HXP + \Pi + P$ is strongly complete for the class of concrete hybrid data models satisfying the frame conditions $FC(\Pi) \wedge FC(P)$.*

3.4. Completeness for path formulas. The main contribution of this article is a characterization of (local) semantic consequence between node formulas by means of an axiomatic system. Given that soundness of an axiomatic system is usually granted, this is the main outcome of the strong completeness result we just proved. In the setting of XPath it also makes sense to discuss the issue of inference between path formulas. Path inference can be traced back to [Pra91], in the context of dynamic algebras. We will now show how Theorem 3.23 can also be used to characterize path inference.

First, recall that previous work like [tCLM10, ADF17] provided *equational* axiomatizations that characterize theorems of the form $\vdash \varphi \equiv \psi$ (for φ, ψ node expressions) and $\vdash \alpha \equiv \beta$ (for α, β path expressions). The first are obviously covered by our results that characterize theoremhood for arbitrary node expressions (in this case \equiv is nothing more than \leftrightarrow). For equivalence between path formulas the following proposition suffices:

Proposition 3.25. *Let α, β be path expressions in $HXPath_{=}$, then $\models \alpha \equiv \beta$ iff $\models @_i \langle \alpha \rangle j \leftrightarrow @_i \langle \beta \rangle j$ for i, j not appearing in α, β .*

Hence, completeness for equivalence theorems between path formulas follow from Theorem 3.23. More interesting is to consider whether a strong completeness result for path consequence is also possible. We need first to introduce this notion. There exists at least two natural possible definitions:

Definition 3.26. Let $\Lambda \cup \{\alpha\}$ be a set of path formulas in $HXPath_{=}$, and let \mathfrak{C} be a class of models. Define

- $\Lambda \models_{\mathfrak{C}}^1 \alpha$ iff for any model $\mathcal{M} \in \mathfrak{C}$, $(\forall m, n, \mathcal{M}, m, n \models \Lambda$ implies $\mathcal{M}, m, n \models \alpha)$.
- $\Lambda \models_{\mathfrak{C}}^2 \alpha$ iff for any model $\mathcal{M} \in \mathfrak{C}$, $(\forall m, n, \mathcal{M}, m, n \models \Lambda$ implies $\forall m, n, \mathcal{M}, m, n \models \alpha)$.

As before we write \models instead of $\models_{\mathfrak{C}}$ if the intended class is clear from context. From the above definition it is easy to show that $\models_{\mathfrak{C}}^1$ implies $\models_{\mathfrak{C}}^2$ but not vice versa. Now, both \models^1 and \models^2 can be captured using Theorem 3.23.

Theorem 3.27. *Let Π be a set of pure axioms and P be a set of existential saturation rules. Let $\mathfrak{C} = \text{Mod}(\mathfrak{F})$ for \mathfrak{F} the class of frames satisfying $FC(\Pi) \wedge FC(P)$. Let $\Lambda \cup \{\alpha\}$ be a set of path expressions in $HXPath_{=}$ then*

- (1) $\Lambda \models_{\mathfrak{C}}^1 \alpha$ implies $\{ @_i \langle \beta \rangle j \mid \beta \in \Lambda \} \vdash @_i \langle \alpha \rangle j$, for i, j not in $\Lambda \cup \{\alpha\}$.
- (2) $\Lambda \models_{\mathfrak{C}}^2 \alpha$ implies $\{ @_i \langle \beta \rangle j \mid \beta \in \Lambda \} \vdash @_k \langle \alpha \rangle l$, for i, j, k, l not in $\Lambda \cup \{\alpha\}$.

Proof. The result is a corollary of Theorem 3.23 because the following hold:

- (1) $\Lambda \models_{\mathfrak{C}}^1 \alpha$ iff $\{ @_i \langle \beta \rangle j \mid \beta \in \Lambda \} \models_{\mathfrak{C}} @_i \langle \alpha \rangle j$, for i, j not in $\Lambda \cup \{\alpha\}$.
- (2) $\Lambda \models_{\mathfrak{C}}^2 \alpha$ iff $\{ @_i \langle \beta \rangle j \mid \beta \in \Lambda \} \models_{\mathfrak{C}} @_k \langle \alpha \rangle l$, for i, j, k, l not in $\Lambda \cup \{\alpha\}$. □

3.5. Some Concrete Examples of Pure Extensions. In this section we introduce some extensions of $HXPath_{=}$, and their corresponding extended axiomatic systems. In all cases, we can apply the result from previous section and automatically obtain completeness, since we only use pure axioms and/or existential saturation rules.

Backwards Navigation. We start with the language $\text{HXPath}_=(-)$, i.e., $\text{HXPath}_=$ extended with the path expression \mathbf{a}^- (backwards navigation). The intuitive semantics for \mathbf{a}^- is

$$\mathcal{M}, x, y \models \mathbf{a}^- \text{ iff } xR_{\mathbf{a}}^{-1}y,$$

i.e., \mathbf{a}^- is interpreted on the inverse of the accessibility relation associated to \mathbf{a} . Equivalently,

$$\mathcal{M}, x, y \models \mathbf{a}^- \text{ iff } yR_{\mathbf{a}}x.$$

In fact, following our presentation, \mathbf{a}^- is an additional modal operator from Mod (to which all rules and axioms of HXP apply), and in addition we will insist that in models of $\text{HXPath}_=(-)$, this accessibility relation is always interpreted as the inverse of the one for \mathbf{a} .

Formally, consider two modal symbols \mathbf{a} and \mathbf{a}^- in Mod (and assume that their respective accessibility relations in a model are $R_{\mathbf{a}}$ and $R_{\mathbf{a}^-}$). Define the set Π_1 of pure axioms that characterizes the interaction between \mathbf{a} and \mathbf{a}^- as

$$\begin{array}{c} \hline \textbf{Axioms for } \mathbf{a}, \mathbf{a}^- \text{-interaction} \\ \hline \text{down-up } \vdash i \rightarrow [\mathbf{a}]\langle \mathbf{a}^- \rangle i \\ \text{up-down } \vdash i \rightarrow [\mathbf{a}^-]\langle \mathbf{a} \rangle i \\ \hline \end{array}$$

Since the axioms presented above are pure, the axiom system we obtain is complete for models obtained from frames satisfying $\text{FC}(\Pi_1)$. It is a simple exercise to verify that $\text{FC}(\Pi_1)$ is equivalent to $\forall x.\forall y.(R_{\mathbf{a}}(x, y) \leftrightarrow R_{\mathbf{a}^-}(y, x))$.

Proposition 3.28. *HXP + Π_1 is sound and strongly complete for $\text{HXPath}_=(-)$.*

Sibling Navigation. We consider now the extension of $\text{HXPath}_=(-)$ with sibling navigation \mathbf{s} , denoted $\text{HXPath}_=(-, \mathbf{s})$. Intuitively,

$$\mathcal{M}, x, y \models \mathbf{s} \text{ iff } x \neq y \text{ and there is some } z \text{ s.t. } zR_{\mathbf{a}}x \text{ and } zR_{\mathbf{a}}y,$$

where $\mathbf{a} \in \text{Mod}$ is fixed.

Consider now three modal symbols \mathbf{a} , \mathbf{a}^- and \mathbf{s} in Mod (and their respective accessibility relations $R_{\mathbf{a}}$, $R_{\mathbf{a}^-}$ and $R_{\mathbf{s}}$). Define the set Π_2 of pure axioms that characterizes their interaction as the axioms in Π_1 together with

$$\begin{array}{c} \hline \textbf{Axioms for siblings} \\ \hline \text{is-sib } \vdash \langle \mathbf{s} \rangle i \rightarrow \langle \mathbf{a}^- \rangle \langle \mathbf{a} \rangle i \\ \text{has-sib } \vdash \neg i \wedge \langle \mathbf{a}^- \rangle \langle \mathbf{a} \rangle i \rightarrow \langle \mathbf{s} \rangle i \\ \text{irref-sib } \vdash i \rightarrow \neg \langle \mathbf{s} \rangle i \\ \hline \end{array}$$

As Π_2 extends Π_1 , $\text{FC}(\Pi_2)$ ensures that $R_{\mathbf{a}^-}$ and $R_{\mathbf{a}}$ are inverses, and in addition that $\forall x.\forall y.(R_{\mathbf{s}}(x, y) \leftrightarrow ((x \neq y) \wedge (\exists z.(R_{\mathbf{a}^-}(x, z) \wedge R_{\mathbf{a}}(z, y))))$.

Proposition 3.29. *HXP + Π_2 is sound and strongly complete for $\text{HXPath}_=(-, \mathbf{s})$.*

Data Equality Properties. As a final, very simple, example let us consider pure axioms defining the behaviour of equality tests. Consider a language with two equality test operators $=_e$ and $=_d$ such that one defines finer equivalence classes than the other (i.e., if \sim_e and \sim_d are their respective accessibility relations we want to ensure that $\sim_e \subseteq \sim_d$). Let us call this logic $\text{HXPath}_=(\subseteq)$. Let Π_3 be the singleton set containing the axiom

Axioms for equality inclusion

$$\text{incl} \vdash \langle @_i =_e @_j \rangle \rightarrow \langle @_i =_d @_j \rangle$$

$\text{FC}(\Pi_3)$ is equivalent to $\forall x.\forall y.(D_e(x, y) \rightarrow D_d(x, y))$.

Proposition 3.30. *HXP + Π_3 is sound and strongly complete for $\text{HXPath}_=(\subseteq)$.*

4. HYBRID XPATH ON TREES

The Henkin-style model construction from Section 3.3 provides us with the tools to obtain complete axiomatizations for a wide variety of extensions of $\text{HXPath}_=$. However there are interesting cases in which completeness is not a direct corollary of the results we already presented. One such case is the class of tree models which we call \mathfrak{C}_{tree} .

We will devote this section to investigate hybrid XPath over the class of tree models. First, in Section 4.1 we will show that it is not possible to get a strongly complete finitary first-order axiomatization over trees. In Section 4.2 we show that it is possible to obtain a strongly complete infinite first-order axiomatization over a slightly larger class of models, while in Section 4.3 we prove that this axiomatization is weakly complete for \mathfrak{C}_{tree} .

Tree models are interesting in this context since XPath is commonly used as a query language over XML documents; and mathematically, the relational structure of an XML document is a tree. In our setting, this consists of asking that the union of all relations in a model has a tree-shape.

Without loss of generality, in the rest of this section we will work in a mono-modal setting, with a unique basic path expression \mathbf{a} , and with single equality/inequality comparisons denoted $=$ and \neq . The path expression \mathbf{a} will be interpreted over an accessibility relation $R_{\mathbf{a}}$, representing the union of all the accessibility relations in the model. To achieve this, we need to work in a signature where the set Mod is finite. Therefore, we can include the following additional axiom:

$$\text{a-definition} \quad \langle \mathbf{a} \rangle \top \leftrightarrow \langle \mathbf{a}_1 \cup \dots \cup \mathbf{a}_k \rangle \top \quad \text{with } \text{Mod} = \{\mathbf{a}_1, \dots, \mathbf{a}_k\}$$

It is a straightforward exercise to show that this axiom characterizes $R_{\mathbf{a}}$ as the union of all the relations of the model, in a signature where Mod is finite. As a consequence, in this section we will focus on axiomatizing $R_{\mathbf{a}}$ as the accessibility relation of a tree. In this way, we obtain the intended meaning for XPath as a query language for XML documents.

Hence, a model in this signature is a tuple $\mathcal{M} = \langle M, \sim, R_{\mathbf{a}}, V, \text{nom} \rangle$, in which we have a unique data equality relation \sim and the accessibility relation $R_{\mathbf{a}}$.

4.1. Failure of Strong Completeness over Trees. Let us start by formally defining the structures we will consider.

Definition 4.1. \mathfrak{C}_{tree} is defined as the class of abstract hybrid data models $\langle M, \sim, R_{\mathbf{a}}, V, \text{nom} \rangle$ satisfying:

- there is a unique point in M without predecessors by $R_{\mathbf{a}}$, which we call the *root*;
- every node $n \in M$ is reachable from the root by $R_{\mathbf{a}}$ in zero or more steps;
- for all $n, l, l' \in M$, if $lR_{\mathbf{a}}n$ and $l'R_{\mathbf{a}}n$, then $l = l'$; and
- there is no $n \in M$ such that n is reachable by $R_{\mathbf{a}}$ from itself in one or more steps.

Intuitively, a model $\langle M, \sim, R_a, V, nom \rangle$ is in \mathfrak{C}_{tree} if $\langle M, R_a \rangle$ is a tree.

In order to show that there is no finitary first-order axiomatization for $HXPath_=$ over \mathfrak{C}_{tree} , we will use a standard tool from first-order model theory: the *compactness* theorem. In particular, we will show that $HXPath_=$ over \mathfrak{C}_{tree} is not compact, and consequently, a finitary axiomatization may not exist (see, e.g., [End01] for details).

Proposition 4.2. *$HXPath_=$ is not compact over the class \mathfrak{C}_{tree} .*

Proof. We need to show that there exists an infinite set of $HXPath_=$ -formulas such that every finite subset $\Gamma^{fin} \subset \Gamma$ is satisfiable in the class \mathfrak{C}_{tree} , but Γ is not.

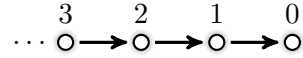
Without loss of generality, we will use natural numbers as names for nominals, i.e., $Nom = \mathbb{N}$. We start by defining the following formula indicating that, at the given point, the nominal k is the only one satisfied from the set $\{0, \dots, n\}$:

$$\text{Only}_k^n = @_k \left(\bigwedge_{0 \leq j \neq k \leq n} \neg j \right).$$

We can define for $n \geq 0$:

$$\text{Lin}(n) = \bigwedge_{0 < i \leq n} @_i(\langle a \rangle i - 1) \wedge \bigwedge_{0 \leq i \leq n} \text{Only}_i^n.$$

The formula $\text{Lin}(n)$ states that there exists a linear chain of named states of length n . Now let us define the set $\Gamma_{\text{Lin}} = \{\text{Lin}(n) \mid n \in \mathbb{N}\}$. Notice that Γ_{Lin} enforces structures with at least one infinite branch of the shape



Γ_{Lin} satisfies the following properties:

- (1) Γ_{Lin} does not have a model in \mathfrak{C}_{tree} , since it enforces an infinite chain without a root (since every node has a predecessor), and
- (2) for all finite sets $\Gamma_{\text{Lin}}^{fin} \subset \Gamma_{\text{Lin}}$, $\Gamma_{\text{Lin}}^{fin}$ has a model in \mathfrak{C}_{tree} .

Hence, the proposition follows. \square

Theorem 4.3. *There is no finitary first-order axiomatization which is strongly complete for $HXPath_=$ over \mathfrak{C}_{tree} .*

Proof. Suppose there is some finitary first-order axiomatization H which is sound and strongly complete for $HXPath_=$ over \mathfrak{C}_{tree} , and let \vdash be obtained from H . For any set of formulas Γ , we know:

- (\dagger) if for all finite set $\Gamma^{fin} \subseteq \Gamma$, Γ^{fin} is consistent (i.e., $\Gamma^{fin} \not\vdash \perp$), then Γ is consistent.

This follows from the fact that any proof is finite, so in order to make Γ inconsistent, only a finite set of its formulas is needed.

Since H is strongly complete, by item 2 in Proposition 4.2, each set $\Gamma_{\text{Lin}}^{fin}$ defined therein is consistent. Then, by item (\dagger) also Γ_{Lin} is consistent, and again by strong completeness of H we can conclude that Γ_{Lin} has a model in \mathfrak{C}_{tree} , contradicting item 1 in Proposition 4.2. Therefore, there is no finitary first-order axiomatic system H strongly complete for \mathfrak{C}_{tree} . \square

Corollary 4.4. *There are no Π and P such that $HXP + \Pi + P$ is strongly complete for \mathfrak{C}_{tree} .*

4.2. Axiomatizing the Class \mathfrak{C}_{forest^-} with Pure Axioms. Define the class \mathfrak{C}_{forest^-} as follows:

Definition 4.5. Let \mathfrak{C}_{forest^-} be the class of models $\langle M, \sim, R_a, V, nom \rangle$ such that

- there is no $n \in M$ such that n is reachable by R_a from itself in one or more steps;
- for all $n \in M$, n has at most one predecessor by R_a .

Notice that the class \mathfrak{C}_{forest^-} admits forests made of a collection of models that consist of a tree, possibly extended with an infinite chain attached to the root. Models from the class \mathfrak{C}_{forest^-} differ from those in \mathfrak{C}_{tree} , given that we relaxed the first two conditions from Definition 4.1. We will introduce a strongly complete axiomatization for \mathfrak{C}_{forest^-} . Let Π_{forest^-} be the set of axioms below. Define a^1 as a , and a^{n+1} as aa^n .

Axioms for forests

$$\begin{array}{l} no-loops \vdash @_i \neg \langle a^n \rangle i \quad \text{for all } n > 0 \\ no-join \vdash @_j \langle a \rangle i \wedge @_k \langle a \rangle i \rightarrow @_j k \end{array}$$

In the table above, *no-loops* is an infinite set of axioms preventing loops of any size, whereas *no-join* prevents the existence of more than one predecessor. Since our axioms are pure and enforce the appropriate structures, from Theorem 3.23 we get:

Theorem 4.6. *The axiomatic system $HXP + \Pi_{forest^-}$ is strongly complete for \mathfrak{C}_{forest^-} .*

4.3. A Weakly Complete Axiomatization for Trees. In this section we will show that the axiom system $HXP + \Pi_{forest^-}$ introduced before, is weakly complete for \mathfrak{C}_{tree} . Recall that for *weak completeness* we need to show that for any HXPath₌-formula φ , $\models \varphi$ implies $\vdash \varphi$. In order to achieve this result we need to work on the extracted model.

In what follows we use $mR_a^h n$ to denote $mR_a u_1 R_a \dots R_a u_{h-1} R_a n$, for some sequence of states u_1, \dots, u_{h-1} .

Definition 4.7. Let $\mathcal{M} = \langle M, \sim, R_a, V, nom \rangle$ be a hybrid data model, $N \subseteq \text{Nom}$, $N \neq \emptyset$, and $n \geq 0$. We define $\mathcal{M}_{\upharpoonright(n,N)}$ as

$$\mathcal{M}_{\upharpoonright(n,N)} = \langle S, \sim_{\upharpoonright S}, R_{a\upharpoonright S}, V_{\upharpoonright S}, nom_{\upharpoonright S} \rangle,$$

where

- $S = \{m \mid \text{there is some } i \in N \text{ such that } nom(i) \rightarrow^h m, \text{ with } h \leq n\}$,
- $\sim_{\upharpoonright S} = \sim \cap (S \times S)$;
- $R_{a\upharpoonright S} = R_a \cap (S \times S)$;
- $V_{\upharpoonright S}(m) = V(m)$, for all $m \in S$; and
- $nom_{\upharpoonright S}(i) = \begin{cases} nom(i) & \text{if } i \in N \\ m & \text{if } i \notin N, m \in S \text{ arbitrary.} \end{cases}$

Intuitively, $\mathcal{M}_{\upharpoonright(n,N)}$ is the restriction of \mathcal{M} to the states reached from a nominal in N , in at most n steps. It is obvious that $\mathcal{M}_{\upharpoonright(n,N)}$ has finite depth if N is finite.

In what follows, we will write $\text{Nom}(\varphi)$ to denote the set of nominals appearing in the formula φ .

Proposition 4.8. *Let φ be an HXPath₌-formula, and let $\mathcal{M}_\Gamma = \langle M, \sim, R_a, V, nom \rangle$ be the extracted model from some $HXP + \Pi_{forest^-}$ MCS Γ , such that $\mathcal{M}_\Gamma, \Delta_i \models \varphi$, for some Δ_i . Let $(\mathcal{M}_\Gamma)_{\upharpoonright(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})} = \langle S, \sim_{\upharpoonright S}, R_{a\upharpoonright S}, V_{\upharpoonright S}, nom_{\upharpoonright S} \rangle$, then,*

- (1) $\Delta_i \in (\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}$, and
(2) $(\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}, \Delta_i \models \varphi$.

Proof. Item 1 follows from the definition of $(\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}$, since $i \in \Delta_i$ by MCS. For item 2, we will prove a stronger result.

(IH1): For all $\alpha \in \text{PExp}$ such that $\text{md}(\alpha) \leq \text{md}(\varphi)$ and $\text{Nom}(\alpha) \subseteq \text{Nom}(\varphi)$, if $\Delta_j, \Delta_k \in (\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}$ then

$$\mathcal{M}_\Gamma, \Delta_j, \Delta_k \models \alpha \text{ iff } (\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}, \Delta_j, \Delta_k \models \alpha.$$

(IH2): For all $\psi \in \text{NExp}$ such that $\text{md}(\psi) \leq \text{md}(\varphi)$ and $\text{Nom}(\psi) \subseteq \text{Nom}(\varphi)$, if $\Delta_j \in (\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}$ then

$$\mathcal{M}_\Gamma, \Delta_j \models \psi \text{ iff } (\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}, \Delta_j \models \psi.$$

The proof is by induction on α and ψ . Assume $\Delta_j, \Delta_k \in (\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}$. Let us start by the base cases.

- $\alpha = \mathbf{a}$: Suppose $\mathcal{M}_\Gamma, \Delta_j, \Delta_k \models \mathbf{a}$, iff $\Delta_j R_a \Delta_k$. Since $\Delta_j, \Delta_k \in (\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}$, by definition we have $\Delta_j \rightarrow_{\uparrow S} \Delta_k$. Therefore, $(\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}, \Delta_j, \Delta_k \models \mathbf{a}$. The other direction is straightforward, since $\Delta_j R_a \uparrow_S \Delta_k$ implies $\Delta_j R_a \Delta_k$.
- $\alpha = @_l$: Suppose $\mathcal{M}_\Gamma, \Delta_j, \Delta_k \models @_l$, iff $\text{nom}(l) = \Delta_k$. By assumption, $\text{Nom}(@_l) \subseteq \text{Nom}(\varphi)$, then $\text{nom}_{\uparrow S}(l) = \text{nom}(l) = \Delta_k$, therefore $(\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}, \Delta_j, \Delta_k \models @_l$. The other direction follows from the fact that $\text{Nom}(@_l) = \{l\} \subseteq \text{Nom}(\varphi)$, then $\text{nom}(l) = \Delta_k$.
- $\psi = p$: Suppose $\mathcal{M}_\Gamma, \Delta_j \models p$, iff $p \in V(\Delta_j)$. By assumption $\Delta_j \in S$, then $p \in V_{\uparrow S}(\Delta_j)$. Hence, $(\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}, \Delta_j \models p$. The other direction is similar.
- $\psi = k \in \text{Nom}(\varphi)$: Suppose $\mathcal{M}_\Gamma, \Delta_j \models k$, iff $\text{nom}(k) = \Delta_j$. By assumption $\Delta_j \in S$, then $\text{nom}_{\uparrow S} = \Delta_j$. Hence, $(\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}, \Delta_j \models k$. The other direction is similar.

Now we prove the inductive cases.

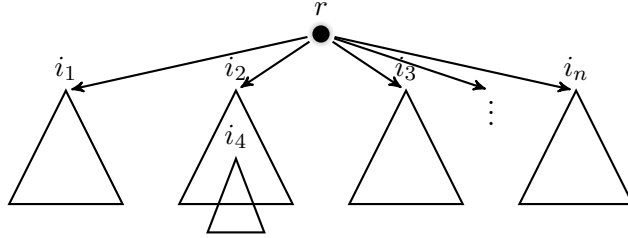
- $\alpha = [\theta]$: Suppose $\mathcal{M}_\Gamma, \Delta_j, \Delta_k \models [\theta]$, iff $\Delta_j = \Delta_k$ and $\mathcal{M}_\Gamma, \Delta_k \models \theta$. By $\Delta_j \in S$ and (IH2), we have $(\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}, \Delta_j \models \theta$, iff $(\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}, \Delta_j, \Delta_k \models [\theta]$. For the other direction, use similar steps.
- $\alpha = \beta\gamma$: Suppose $\mathcal{M}_\Gamma, \Delta_j, \Delta_k \models \beta\gamma$, iff there exists Δ_l such that $\mathcal{M}_\Gamma, \Delta_j, \Delta_l \models \beta$ and $\mathcal{M}_\Gamma, \Delta_l, \Delta_k \models \gamma$. Notice that since $\Delta_j, \Delta_k \in S$, and $\text{md}(\alpha) \leq \text{md}(\varphi)$, we have $\Delta_j \rightarrow^n \Delta_l$, for some $n \leq \text{md}(\varphi)$, then $\Delta_l \in S$. By (IH1), we have $(\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}, \Delta_j, \Delta_l \models \beta$ and $(\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}, \Delta_l, \Delta_k \models \gamma$. Therefore, $(\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}, \Delta_j, \Delta_k \models \beta\gamma$. The other direction is direct from (IH1).
- $\psi = \neg\theta$ and $\psi = \theta \wedge \theta'$: are direct from (IH2).
- $\psi = \langle \beta = \gamma \rangle$: $\mathcal{M}_\Gamma, \Delta_j \models \langle \beta = \gamma \rangle$, iff there exists Δ_k, Δ_l such that $\mathcal{M}_\Gamma, \Delta_j, \Delta_k \models \beta$, $\mathcal{M}_\Gamma, \Delta_j, \Delta_l \models \gamma$ and $\Delta_k \sim \Delta_l$. From the fact that $\text{md}(\psi) \leq \text{md}(\varphi)$, we have $\text{md}(\beta) \leq \text{md}(\varphi)$ and $\text{md}(\gamma) \leq \text{md}(\varphi)$. Then, $\Delta_j \rightarrow^n \Delta_k$, and $\Delta_j \rightarrow^m \Delta_l$, for some $n, m \leq \text{md}(\varphi)$. As a consequence, $\Delta_k, \Delta_l \in S$, then by (IH1) we get $(\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}, \Delta_j, \Delta_k \models \beta$ and $(\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}, \Delta_j, \Delta_l \models \gamma$. On the other hand, since $\Delta_k \sim \Delta_l$, we also get $\Delta_k \sim_{\uparrow S} \Delta_l$, therefore $(\mathcal{M}_\Gamma)_{\uparrow(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}, \Delta_j \models \langle \beta = \gamma \rangle$. The other direction follows from (IH1).
- $\psi = \langle \beta \neq \gamma \rangle$: Analogous to the previous case. □

With this construction at hand, we obtain almost the structure of the model we are looking for.

Lemma 4.9. *Let \mathcal{M}_Γ be the extracted model for some MCS Γ , $n \in \mathbb{N}$ and $N \subseteq \text{Nom}$ finite. $(\mathcal{M}_\Gamma)_{\upharpoonright(n,N)}$ is a forest composed by a finite number of disjoint trees.*

Proof. It is a straightforward exercise to check that from each nominal $i \in N$, we generate a tree (remember that by *no-loops* there are no cycles in \mathcal{M}_Γ , and by *no-join* each node has at most one predecessor). Since N is finite, we generate only a finite number of trees. \square

Definition 4.10. Let $\mathcal{M}_{\upharpoonright(n,N)}$ be a model as in Definition 4.7. We define $\mathcal{M}_{\upharpoonright(n,N)}^{IT}$ as the tree resulting of adding a new node r to $\mathcal{M}_{\upharpoonright(n,N)}$, and edges from r to the root to each of its subtrees. Essentially, $\mathcal{M}_{\upharpoonright(n,N)}^{IT}$ is of the form:



Proposition 4.11. *Let \mathcal{M}_Γ be the extracted model for some $\text{HXP} + \Pi_{\text{forest-}}$ MCS Γ and let φ be an $\text{HXPath}_=$ -formula such that $\mathcal{M}_\Gamma, \Delta_i \models \varphi$, for some Δ_i . Then,*

- (1) $(\mathcal{M}_\Gamma)_{\upharpoonright(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}^{IT}$ is a tree of finite height, and
- (2) $(\mathcal{M}_\Gamma)_{\upharpoonright(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}^{IT}, \Delta_i \models \varphi$.

Proof. For 1, by construction it is obvious that $(\mathcal{M}_\Gamma)_{\upharpoonright(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}^{IT}$ is a tree of finite height. To prove 2 we can use a bisimulation-based argument. Define Z as the identity relation between the states in $(\mathcal{M}_\Gamma)_{\upharpoonright(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}$ and $(\mathcal{M}_\Gamma)_{\upharpoonright(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}^{IT}$ (therefore r is not in the relation). It is a straightforward exercise to show that Z is an $\text{HXPath}_=$ -bisimulation. Since $(\mathcal{M}_\Gamma)_{\upharpoonright(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}, \Delta_i \models \varphi$, by Proposition 2.11 we get $(\mathcal{M}_\Gamma)_{\upharpoonright(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}^{IT}, \Delta_i \models \varphi$. \square

Because we were able to transform the extracted model into a model in $\mathfrak{C}_{\text{tree}}$ which still satisfies the intended formula, we obtain the intended completeness result.

Theorem 4.12. *The system $\text{HXP} + \Pi_{\text{forest-}}$ is weakly complete for $\mathfrak{C}_{\text{tree}}$.*

Moreover, $(\mathcal{M}_\Gamma)_{\upharpoonright(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}^{IT}$ can be transformed into a bounded finite model, and hence, also decidability of the validity problem follows.

Definition 4.13. Let φ be an $\text{HXPath}_=$ -formula, and let $\mathcal{M} = \langle M, \sim, R_a, V, \text{nom} \rangle \in \mathfrak{C}_{\text{tree}}$. We write $m \equiv_\Gamma m'$ if for all n , $nR_a m$ if and only if $nR_a m'$, and $\mathcal{M}, m \models \varphi$ if and only if $\mathcal{M}, m' \models \varphi$, for all $\varphi \in \Gamma$. If Γ is finite, \equiv_Γ is an equivalence relation of finite index.

Given an equivalence relation \equiv over a set M , a *selection function* s for \equiv is a function $s : M_{/\equiv} \rightarrow M$ such that $s([m]) = m'$, for $m' \in [m]$ arbitrary.

Definition 4.14. Let $(\mathcal{M}_\Gamma)_{\upharpoonright(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}^{IT} = \langle M, \sim, R_a, V, \text{nom} \rangle$ be as in Definition 4.10, let φ be an $\text{HXPath}_=$ -formula, and let s be a selection function for $\equiv_{\text{Sub}(\varphi)}$. We define $(\mathcal{M}_\Gamma)_{\upharpoonright(\text{md}(\varphi), \text{Nom}(\varphi) \cup \{i\})}^{FT} = \langle F, \sim_{\upharpoonright F}, R_a_{\upharpoonright F}, V_{\upharpoonright F}, \text{nom}_{\upharpoonright F} \rangle$ where

- $F = \{s([m]) \mid m \in M\}$;

- $\sim_{\uparrow F} = \sim \cap (F \times F)$;
- $R_{\mathbf{a}} \uparrow F = R_{\mathbf{a}} \cap (F \times F)$;
- $V_{\uparrow F}(m) = V(m)$, for all $m \in F$; and
- $nom_{\uparrow F}(i) = \begin{cases} nom(i) & \text{if } i \in \mathbf{Nom}(\varphi) \\ m & \text{if } i \notin \mathbf{Nom}(\varphi), m \in F \text{ arbitrary.} \end{cases}$

Proposition 4.15. *Let \mathcal{M}_{Γ} be the extracted model for some $\text{HXP} + \Pi_{\text{forest-}} \text{MCS } \Gamma$ and let φ be an $\text{HXPath}_{=}$ -formula such that $\mathcal{M}_{\Gamma}, \Delta_i \models \varphi$, for some Δ_i . Then,*

- (1) $(\mathcal{M}_{\Gamma})_{\uparrow(\text{md}(\varphi), \mathbf{Nom}(\varphi) \cup \{i\})}^{FT}$ is a finite tree, and
- (2) $(\mathcal{M}_{\Gamma})_{\uparrow(\text{md}(\varphi), \mathbf{Nom}(\varphi) \cup \{i\})}^{FT}, s([\Delta_i]) \models \varphi$.

Proof. 1 follows from the fact that $(\mathcal{M}_{\Gamma})_{\uparrow(\text{md}(\varphi), \mathbf{Nom}(\varphi) \cup \{i\})}^{IT}$ is a tree of finite height, and the selection function selects only one representant for each equivalence class, which gives us finite width. In order to prove 2, let $d(m)$ be the distance from m to the root, define:

$$mZ_{\text{md}(\varphi)} s([m]) \quad \text{if } d(m) \leq d(\Delta_i)$$

$$mZ_{\max(\text{md}(\varphi) - (d(\Delta_i) - d(m)), 0)} s([m]) \quad \text{if } d(m) > d(\Delta_i).$$

We claim

$$(\mathcal{M}_{\Gamma})_{\uparrow(\text{md}(\varphi), \mathbf{Nom}(\varphi) \cup \{i\})}^{IT}, \Delta_i \stackrel{\text{md}(\varphi)}{\Leftrightarrow} (\mathcal{M}_{\Gamma})_{\uparrow(\text{md}(\varphi), \mathbf{Nom}(\varphi) \cup \{i\})}^{FT}, s([\Delta_i]),$$

therefore (by Propositions 2.13 and 4.11), $(\mathcal{M}_{\Gamma})_{\uparrow(\text{md}(\varphi), \mathbf{Nom}(\varphi) \cup \{i\})}^{FT}, s([\Delta_i]) \models \varphi$. \square

As a consequence we get:

Theorem 4.16. *$\text{HXPath}_{=}$ on trees has the bounded model property.*

The model constructed in Definition 4.15 has size at most exponential in the input formula. Moreover, the algorithm below shows that model checking for $\text{HXPath}_{=}$ can be solved in polynomial time. Thus, for a given formula we can guess a model of exponential size and do model checking in polynomial time, solving the problem in NEXPSpace (which coincides with the class EXPSpace).

Let $\mathcal{M} = \langle M, \{\sim_e\}_{e \in \text{Eq}}, \{R_{\mathbf{a}}\}_{\mathbf{a} \in \text{Mod}}, V, nom \rangle$ be a model and φ a formula in $\text{HXPath}_{=}$, let e_1, \dots, e_n be an enumeration, ordered by size, of the subexpressions of φ . Define the functions $Label_{ne}$ and $Label_{pe}$ inductively as follows. Initially, set $Label_{ne}(m) = Label_{pe}(m_1, m_2) = \{\}$ for any m, m_1, m_2 . Then extend these labeling functions using the following rules.

- (1) $e_i = \mathbf{a} \in \text{Mod}$: $add(Label_{pe}(m_1, m_2), \mathbf{a})$ iff $R_{\mathbf{a}}(m_1, m_2)$.
- (2) $e_i = @_i$: $add(Label_{pe}(m_1, m_2), @_i)$ iff $m_2 = nom(i)$.
- (3) $e_i = [\psi]$: $add(Label_{pe}(m, m), [\psi])$ iff $m \in Label_{ne}(\psi)$.
- (4) $e_i = \mathbf{a}\alpha$: $add(Label_{pe}(m_1, m_2), \mathbf{a}\alpha)$ iff there is m_3 s.t. $R_{\mathbf{a}}(m_1, m_3)$ and $\alpha \in Label_{pe}(m_3, m_2)$.
- (5) $e_i = @_i\alpha$: $add(Label_{pe}(m_1, m_2), @_i\alpha)$ iff $\alpha \in Label_{pe}(nom(i), m_2)$.
- (6) $e_i = [\psi]\alpha$: $add(Label_{pe}(m_1, m_2), [\psi]\alpha)$ iff $\psi \in Label_{ne}(m_1)$ and $\alpha \in Label_{pe}(m_1, m_2)$.
- (7) $e_i = p \in \text{Prop}$: $add(Label_{ne}(m), p)$ iff $p \in V(m)$.
- (8) $e_i = i \in \text{Nom}$: $add(Label_{ne}(m), i)$ iff $m = nom(i)$.
- (9) $e_i = \neg\psi$: $add(Label_{ne}(m), \neg\psi)$ iff $\psi \notin Label_{ne}(m)$.
- (10) $e_i = \psi_1 \wedge \psi_2$: $add(Label_{ne}(m), \psi_1 \wedge \psi_2)$ iff $\psi_1, \psi_2 \in Label_{ne}(m)$.
- (11) $e_i = \langle \alpha_1 =_e \alpha_2 \rangle$: $add(Label_{ne}(m), \langle \alpha_1 =_e \alpha_2 \rangle)$ iff there are m_1, m_2 s.t. $\alpha_1 \in Label_{pe}(m, m_1)$, $\alpha_2 \in Label_{pe}(m, m_2)$ and $m_1 \sim_e m_2$.
- (12) $e_i = \langle \alpha_1 \neq_e \alpha_2 \rangle$: $add(Label_{ne}(m), \langle \alpha_1 \neq_e \alpha_2 \rangle)$ iff there are m_1, m_2 s.t. $\alpha_1 \in Label_{pe}(m, m_1)$, $\alpha_2 \in Label_{pe}(m, m_2)$ and it is not the case that $m_1 \sim_e m_2$.

As a result we obtain the following upper bound for HXPath₌-satisfiability in \mathfrak{C}_{tree} .

Corollary 4.17. *The satisfiability problem for HXPath₌ on trees is decidable in EXPSPACE.*

5. DECIDABILITY VIA FILTRATIONS

In the previous section we already obtained decidability for HXPath₌ on trees. In this section we introduce *filtrations*, a notion which will help us to establish a bounded finite model property and an upper bound for the complexity of satisfiability for HXPath₌(-)³ over the class of all abstract hybrid data models. Recall that HXPath₌(-) is the extension of HXPath₌ with inverse modalities.

In Section 2 we introduced path expressions of the form $\alpha \cup \beta$ as an abbreviation, but this definition potentially leads to an exponential blow up in the size of the formula. In this section we will consider \cup as part of the language of HXPath₌(-) to avoid this blow up.

Definition 5.1. Let Σ be a set of formulas of HXPath₌. We say that Σ is *closed* if:

- $\neg\varphi \in \Sigma$ implies $\varphi \in \Sigma$
- $\varphi \wedge \psi \in \Sigma$ implies $\varphi, \psi \in \Sigma$
- $\langle \alpha * \beta \rangle \in \Sigma$ implies $\langle \alpha \rangle \top, \langle \beta \rangle \top \in \Sigma$
- $\langle [\varphi] \rangle \top \in \Sigma$ implies $\varphi \in \Sigma$
- $\langle @_i \rangle \top \in \Sigma$ implies $i \in \Sigma$
- $\langle \alpha\beta \rangle \top \in \Sigma$ implies $\langle \alpha \rangle \top, \langle \beta \rangle \top \in \Sigma$
- $\langle \alpha \cup \beta \rangle \top \in \Sigma$ implies $\langle \alpha \rangle \top, \langle \beta \rangle \top \in \Sigma$.

Definition 5.2. Let Σ be a closed set of formulas, and \mathcal{M} be an abstract hybrid data model with m, n states in \mathcal{M} . We define $m \rightsquigarrow_{\Sigma} n$ if and only if for all $\varphi \in \Sigma$ ($\mathcal{M}, m \models \varphi$ iff $\mathcal{M}, n \models \varphi$).

Now let us introduce filtration, the construction that will help us to obtain a *small model property* and give us decidability.

Definition 5.3 (Filtrations). Let $\mathcal{M} = \langle M, \{\sim_e\}_{e \in \text{Eq}}, \{R_a\}_{a \in \text{Mod}}, V, \text{nom} \rangle$, be an abstract hybrid data model, and Σ a closed set of formulas. A *filtration of \mathcal{M} via Σ* is any $\mathcal{M}^f = \langle M^f, \{\sim_e^f\}_{e \in \text{Eq}}, \{R_a^f\}_{a \in \text{Mod}}, V^f, \text{nom}^f \rangle$, such that

- $M^f = M / \rightsquigarrow_{\Sigma}$
- If $m R_a n$ then $[m] R_a^f [n]$
- If $[m] R_a^f [n]$ then for all $\langle a \rangle \top \in \Sigma$, for all $m' \in [m]$ there exists $n' \in [n]$ such that $\mathcal{M}, m', n' \models a$
- $[m] \sim_e^f [n]$ iff $m \sim_e n$
- $\text{nom}^f(i) = [\text{nom}(i)]$
- $V^f([m]) = \{p \mid p \in V(m)\}$.

The *smallest filtration* is a filtration that satisfies

$$[m] R_a^f [n] \text{ iff there exist } m' \in [m], n' \in [n] \text{ such that } m' R_a n'.$$

³In [AFS17] it has been proved that the satisfiability problem for HXPath₌ with a single accessibility relation and a single data relation is PSPACE-complete. However, for multiple relations the exact complexity is unknown.

Theorem 5.4. *Let \mathcal{M} be an abstract hybrid data model and \mathcal{M}^f a filtration of \mathcal{M} for a closed set Σ . Then*

- (1) *For all $\varphi \in \Sigma$, for all $m \in \mathcal{M}$, $\mathcal{M}^f, [m] \models \varphi$ iff $\mathcal{M}, m \models \varphi$, and*
- (2) *For all $\langle \alpha \rangle \top \in \Sigma$, for all $m, n \in \mathcal{M}$, $\mathcal{M}^f, [m], [n] \models \alpha$ iff for all $m' \in [m]$ there exists $n' \in [n]$ such that $\mathcal{M}, m', n' \models \alpha$.*

Proof. By structural induction. First we prove 1. The base and Boolean cases are simple. Let us consider $\langle \alpha =_e \beta \rangle \in \Sigma$. $\mathcal{M}^f, [m] \models \langle \alpha =_e \beta \rangle$ iff there are $[n], [u]$ such that

- (i) $\mathcal{M}^f, [m], [n] \models \alpha$ iff (by IH and $\langle \alpha \rangle \top \in \Sigma$) there exist $m' \in [m], n' \in [n]$ such that $\mathcal{M}, m', n' \models \alpha$,
- (ii) $\mathcal{M}^f, [m], [u] \models \beta$ iff (by IH and $\langle \beta \rangle \top \in \Sigma$) there exist $m'' \in [m], u' \in [u]$ such that $\mathcal{M}, m'', u' \models \beta$, and
- (iii) $[n] \sim_e^f [u]$ iff (definition of \mathcal{M}^f) $n \sim_e u$. Since $[n] = [n']$ and $[u] = [u']$ we have $n' \sim_e u'$.

Notice that the problem is that m' might be different from m'' . Suppose for contradiction that $\mathcal{M}, m', u' \not\models \beta$. Since $\langle \beta \rangle \top \in \Sigma$, $m' \in [m]$, and $u' \in [u]$, by using (ii) above we get $\mathcal{M}^f, [m], [u] \not\models \beta$, which is a contradiction. Therefore, $\mathcal{M}, m' \models \langle \alpha =_e \beta \rangle$. The case for $\langle \alpha \neq_e \beta \rangle$ is similar.

Now let us prove 2:

- Case $\langle [\varphi] \rangle \top \in \Sigma$. For the left to the right direction, suppose $\mathcal{M}^f, [m], [n] \models [\varphi]$, iff $[m] = [n]$ and $\mathcal{M}^f, [m] \models \varphi$. By 1 we have $\mathcal{M}, m \models \varphi$, then by definition of $[m]$, $\mathcal{M}, m' \models \varphi$, for all $m' \in [m]$. Hence $\mathcal{M}, m', m' \models [\varphi]$ as wanted, since $m' \in [n]$.

For the other direction, suppose that for all $m' \in [m]$ there exists $n' \in [n]$ s.t. $\mathcal{M}, m', n' \models [\varphi]$. Then $m' = n'$ and $\mathcal{M}, m' \models \varphi$. By 1, $\mathcal{M}^f, [m'] \models \varphi$, and since $[m] = [m']$, $\mathcal{M}^f, [m] \models \varphi$. Therefore $\mathcal{M}^f, [m], [n] \models [\varphi]$, because $n' = m'$ implies $[n] = [n'] = [m'] = [m]$.

- Case $\langle @_i \rangle \top \in \Sigma$. For the left to the right direction suppose $\mathcal{M}^f, [m], [n] \models @_i$, iff $nom^f(i) = [n]$. By definition of nom^f , we have $nom^f(i) = [nom(i)]$, then $nom(i) = n$. Hence $\mathcal{M}, m, n \models @_i$.

For the other direction, suppose that for all $m' \in [m]$ there exists $n' \in [n]$ s.t. $\mathcal{M}, m', n' \models @_i$, iff $nom(i) = n'$. Then $[n'] = \{n'\} = [n]$, therefore $\mathcal{M}^f, [m], [n] \models @_i$.

- Case $\langle \mathbf{a} \rangle \top \in \Sigma$. For the left to the right direction suppose $\mathcal{M}^f, [m], [n] \models \mathbf{a}$, iff $[m]R_{\mathbf{a}}^f[n]$. Then, since $\langle \mathbf{a} \rangle \top \in \Sigma$, for all $m' \in [m]$ there exists $n' \in [n]$ s.t. $\mathcal{M}, m', n' \models \mathbf{a}$ (by the second condition on the definition of $R_{\mathbf{a}}^f$), as wanted.

For the right to the left direction, suppose for all $m' \in [m]$ there exists $n' \in [n]$ s.t. $\mathcal{M}, m', n' \models \mathbf{a}$. Then $m'R_{\mathbf{a}}n'$, and by the first condition of the definition of $R_{\mathbf{a}}^f$ we have $[m']R_{\mathbf{a}}^f[n]$, iff $[m]R_{\mathbf{a}}^f[n]$. Hence, $\mathcal{M}^f, [m], [n] \models \mathbf{a}$.

- Case $\langle \alpha\beta \rangle \top \in \Sigma$. For the left to the right direction suppose $\mathcal{M}^f, [m], [n] \models \alpha\beta$, iff there exists $[z]$ such that $\mathcal{M}^f, [m], [z] \models \beta$ and $\mathcal{M}^f, [z], [n] \models \alpha\beta$. By IH, for all $m' \in [m]$ there exists $z' \in [z]$ s.t. $\mathcal{M}, m', z' \models \alpha$ and for all $z' \in [z]$ there exists $n' \in [n]$ s.t. $\mathcal{M}, z', n' \models \beta$. This implies that for all $m' \in [m]$ there exists $n' \in [n]$ s.t. $\mathcal{M}, m', n' \models \alpha\beta$.

For the other direction, suppose for all $m' \in [m]$ there exists $n' \in [n]$ s.t. $\mathcal{M}, m', n' \models \alpha\beta$. Then there exists z' s.t. $\mathcal{M}, m', z' \models \alpha$ and $\mathcal{M}, z', n' \models \beta$. By IH we have $\mathcal{M}^f, [m], [z'] \models \alpha$ and $\mathcal{M}^f, [z'], [n] \models \beta$. Hence $\mathcal{M}^f, [m], [n] \models \alpha\beta$.

- Case $\langle \alpha \cup \beta \rangle \top \in \Sigma$. For the left to the right direction suppose $\mathcal{M}^f, [m], [n] \models \alpha \cup \beta$, iff $\mathcal{M}^f, [m], [n] \models \alpha$ or $\mathcal{M}^f, [m], [n] \models \beta$. By IH, for all $m' \in [m]$ there exists $n' \in [n]$ s.t.

$\mathcal{M}, m', n' \models \alpha$ or $\mathcal{M}, m', n' \models \beta$. This implies that for all $m' \in [m]$ there exists $n' \in [n]$ s.t. $\mathcal{M}, m', n' \models \alpha \cup \beta$.

For the other direction, suppose for all $m' \in [m]$ there exists $n' \in [n]$ s.t. $\mathcal{M}, m', n' \models \alpha \cup \beta$, iff $\mathcal{M}, m', n' \models \alpha$ or $\mathcal{M}, m', n' \models \beta$. By IH we have $\mathcal{M}^f, [m], [n] \models \alpha$ or $\mathcal{M}^f, [m], [n] \models \beta$. Hence $\mathcal{M}^f, [m], [n] \models \alpha \cup \beta$. \square

As it happens for many modal logics the previous result gives us a bounded model property for satisfiability: if a formula φ is satisfiable it is satisfiable in a model of size at most exponential in the size of φ . Moreover, filtrations preserve some structural properties of the accessibility relations. In particular the following is true:

Proposition 5.5. *Let $\mathcal{M} = \langle M, \{\sim_e\}_{e \in \text{Eq}}, \{\rightarrow_a\}_{a \in \text{Mod}}, V, \text{nom} \rangle$ be a model, and let \mathcal{M}^f be the smallest filtration of \mathcal{M} . Then, $\mathbf{a}^- = (\mathbf{a})^{-1}$ implies $(\mathbf{a}^-)^f = (\mathbf{a}^f)^{-1}$*

Proof. Suppose that $\mathbf{a}^- = (\mathbf{a})^{-1}$. We want to prove that $(\mathbf{a}^-)^f = (\mathbf{a}^f)^{-1}$. Then suppose $[m]R_{\mathbf{a}^-}^f[n]$. Since f is the smallest filtration, there exist $m' \in [m], n' \in [n]$ such that $m'R_{\mathbf{a}^-}n'$. But by hypothesis, we have $n'R_{\mathbf{a}}m'$, and by definition of filtration we get $[n']R_{\mathbf{a}}^f[m']$. Since $[m'] = [m]$ and $[n'] = [n]$, we obtain $[n]R_{\mathbf{a}}^f[m]$. \square

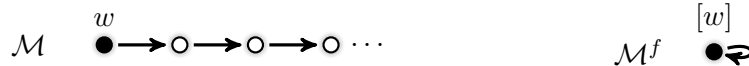
As a result we obtain an upper bound for the complexity of satisfiability.

Corollary 5.6. *The satisfiability problem of $\text{HXPath}_=(\cdot)$ is in NEXPTIME.*

On the other hand, if we consider sibling navigation the argument does not work. In the figure below we represent a relation $R_{\mathbf{a}}$ by black arrows, and the sibling relation $R_{\mathbf{s}}$ by dotted arrows. Let $\Sigma = \{\langle \mathbf{a} \rangle \top\}$. The model on the right is the filtration by Σ of the model on the left. As we can see, any filtration collapses v and u into a unique equivalence class with a reflexive arrow, but the sibling relation must be irreflexive. Hence the notion of filtration we introduced is not appropriate to handle siblings.



Finally, we show that filtration does not work for proving that the satisfiability problem for $\text{HXPath}_=$ over \mathfrak{C}_{tree} is decidable. This is why, in order to show Corollary 4.17 we used a more specialized method. Consider again $\Sigma = \{\langle \mathbf{a} \rangle \top\}$. The model on the right is the filtration by Σ of the model on the left (assuming all nodes are equal for the data relation). Any filtration collapses all nodes from \mathcal{M} into a single reflexive node. Clearly, $\mathcal{M}^f \notin \mathfrak{C}_{tree}$.



6. DISCUSSION

We introduced a sound and strongly complete axiomatization for $\text{HXPath}_=$, i.e., the language XPath with forward navigation for multiple accessibility relations; multiple equality/inequality data comparisons; and where node expressions are extended with nominals, and path expressions are extended with the hybrid operator $@$. The *hybridization* of XPath allowed us to apply a completeness argument similar to the one used for the hybrid logic $\mathcal{HL}(@)$ shown in, e.g., [BtC06]. This ensures that certain extensions of the axiomatic system we introduce are also strongly complete. The axiomatic systems that can be obtained in this way cover a large family of hybrid XPath languages over different classes of frames.

Our system extends the calculus introduced previously in [AF16]. The most important improvement is that we provide a minimal system which can be extended with axioms and rules of certain kind, such that strong completeness immediately follows. The kind of axioms and rules we allow in the extensions ensures completeness with respect to a large family of frame classes. We showed interesting examples of such extensions. In particular, we obtain a strongly complete axiomatization for the logic extended with backward and sibling navigation, and for data equality inclusion.

One particularly interesting class of structures used in practice is the class of *tree models*, since the main applications of XPath in, e.g., web semantics are related to XML documents. From a mathematical point of view, XML documents are trees. In this respect, we investigate axiomatizations for $\text{HXPath}_=$ on tree models. First, we showed that there is no finitary first-order axiomatization which is strongly complete for $\text{HXPath}_=$ on trees. Then we discussed two alternatives for dealing with tree-like structures. On the one hand, it is possible to relax some condition on the models. We consider the class \mathfrak{C}_{forest^-} , which are forest-like models but possibly with infinite chains with respect to the predecessor relation. More precisely, we extend the basic axiom system with an infinite set of pure axioms to strongly axiomatize the class \mathfrak{C}_{forest^-} . Another alternative is giving up *strong* completeness. We showed weak completeness for the class \mathfrak{C}_{tree} , using pure axioms.

We also investigated the status of decidability for the satisfiability problem of $\text{HXPath}_=$ and some extensions. We used a standard technique in modal logics named *filtrations* [BvB06]. The filtration method is a way to build finite models by taking a large model and collapsing as many states as possible. We replicate this technique for $\text{HXPath}_=(-)$, and obtain a NEXPTIME upper bound for its satisfiability problem. We showed that filtrations do not work on some extensions of $\text{HXPath}_=$. In particular, for the case of tree models we proved that the satisfiability problem is decidable in EXPSpace using a more specialized approach.

As future work, it would be interesting to investigate XPath₌ fragments with the reflexive-transitive path \mathbf{a}^* . One of the main limitations of the framework we introduced in this paper is that it can only axiomatize first-order languages. For that reason, transitive closure operators cannot be accounted for, and a different proof strategy is needed. We conjecture that it is possible to adapt the results from [HKT00] for Propositional Dynamic Logic (PDL) to obtain a weakly complete axiom system for $\text{HXPath}_=(*)$ (i.e., $\text{HXPath}_=$ extended with \mathbf{a}^*) over the class of all models. However, for axiomatizing $\text{HXPath}_=(*)$ over trees, the filtration technique used in [HKT00] does not seem to work, and new developments are needed, as for other logics with fix point operators over tree-like structures (see, e.g., the case of CTL* in [Rey01]).

The exact computational complexity of the logics we considered has not been investigated yet. It has been proved that XPath₌ with single accessibility and data relations extended

with hybrid operators is PSPACE-complete [AFS17]. Moreover, we provided upper bounds for multi-modal $HXPath_{=}$ (-) over arbitrary models and for $HXPath_{=}$ on trees, but without giving a tight lower bound. It would be also interesting to investigate the complexity of the multi-modal languages we studied in this article, enriched with backward and sibling navigation, reflexive-transitive closures, and over the class of trees. We conjecture that we can adapt the automata proof given in [Fig10], with the method used to account for hybrid operators presented in [SV01] to get exact bounds.

Acknowledgments. We would like to thank the two anonymous reviewers for their helpful comments and suggestions. This work is supported by projects ANPCyT-PICTs-2017-1130, Stic-AmSud 20-STIC-03 ‘DyLo-MPC’, Secyt-UNC, GRFT Mincyt-Cba, and by the Laboratoire International Associé SINFIN.

REFERENCES

- [ABFF16] S. Abriola, P. Barceló, D. Figueira, and S. Figueira. Bisimulations on data graphs. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR*, pages 309–318, 2016.
- [ABFF18] S. Abriola, P. Barceló, D. Figueira, and S. Figueira. Bisimulations on data graphs. *Journal of Artificial Intelligence Research*, 61:171–213, 2018.
- [ABM01] C. Areces, P. Blackburn, and M. Marx. Hybrid logics: characterization, interpolation and complexity. *The Journal of Symbolic Logic*, 66(3):977–1010, 2001.
- [ADF14] S. Abriola, M. Descotte, and S. Figueira. Definability for downward and vertical XPath on data trees. In *21th Workshop on Logic, Language, Information and Computation*, volume 6642 of *LNCS*, pages 20–34, 2014.
- [ADF17] S. Abriola, M. Descotte, and S. Figueira. Model theory of xpath on data trees. part II: binary bisimulation and definability. *Information and Computation*, 255:195–223, 2017.
- [ADFF17] S. Abriola, M. E. Descotte, R. Fervari, and S. Figueira. Axiomatizations for downward XPath on data trees. *Journal of Computer and System Sciences*, 89:209–245, 2017.
- [AF16] C. Areces and R. Fervari. Hilbert-style axiomatization for hybrid XPath with data. In L. Michael and A. Kakas, editors, *Logics in Artificial Intelligence - 15th European Conference, JELIA 2016, Larnaca, Cyprus, November 9-11, 2016, Proceedings*, volume 10021 of *Lecture Notes in Computer Science*, pages 34–48, 2016.
- [AFL02] M. Arenas, W. Fan, and L. Libkin. On verifying consistency of xml specifications. In *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS ’02, pages 259–270, New York, NY, USA, 2002. ACM.
- [AFS17] C. Areces, R. Fervari, and N. Seiler. Tableaux for hybrid XPath with data. In E. Oliveira, J. Gama, Z. Vale, and H. Lopes Cardoso, editors, *Progress in Artificial Intelligence - 18th EPIA Conference on Artificial Intelligence, EPIA 2017, Porto, Portugal, September 5-8, 2017, Proceedings*, volume 10423 of *Lecture Notes in Computer Science*, pages 611–623. Springer, 2017.
- [AG08] R. Angles and C. Gutiérrez. Survey of graph database models. *ACM Comput. Surv.*, 40(1):1:1–1:39, 2008.
- [BdRV01] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001.
- [BFG08] M. Benedikt, W. Fan, and F. Geerts. XPath satisfiability in the presence of DTDs. *Journal of the ACM*, 55(2):1–79, 2008.
- [BK08] M. Benedikt and C. Koch. XPath leashed. *ACM Computing Surveys*, 41(1), 2008.
- [BLS16] D. Baelde, S. Lunel, and S. Schmitz. A sequent calculus for a modal logic on finite data trees. In *25th EACSL Annual Conference on Computer Science Logic, CSL 2016*, pages 32:1–32:16, 2016.
- [BMSS09] M. Bojańczyk, A. Muscholl, T. Schwentick, and L. Segoufin. Two-variable logic on data trees and XML reasoning. *Journal of the ACM*, 56(3), 2009.
- [Bra11] T. Brauner. *Hybrid Logic and Its Proof-Theory*. Springer, 2011.

- [BtC06] P. Blackburn and B. ten Cate. Pure extensions, proof rules, and hybrid axiomatics. *Studia Logica*, 84(2):277–322, 2006.
- [BvB06] P. Blackburn and J. van Benthem. Modal Logic: A Semantic Perspective. In *Handbook of Modal Logic* [BvBW06], pages 1–84.
- [BvBW06] P. Blackburn, J. van Benthem, and F. Wolter. *Handbook of Modal Logic, Volume 3 (Studies in Logic and Practical Reasoning)*. Elsevier Science Inc., New York, NY, USA, 2006.
- [CD99] J. Clark and S. DeRose. XML path language (XPath). Website, 1999. W3C Recommendation, url = <http://www.w3.org/TR/xpath>.
- [End01] H. Enderton. *A Mathematical Introduction to Logic*. Academic Press, 2nd edition, 2001.
- [FCG04] W. Fan, C. Chan, and M. Garofalakis. Secure XML querying with security views. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, pages 587–598. ACM, 2004.
- [FFA14] D. Figueira, S. Figueira, and C. Areces. Basic model theory of XPath on data trees. In *International Conference on Database Theory*, pages 50–60, 2014.
- [FFA15] D. Figueira, S. Figueira, and C. Areces. Model theory of XPath on data trees. Part I: Bisimulation and characterization. *Journal of Artificial Intelligence Research*, 53:271–314, 2015.
- [Fig10] D. Figueira. *Reasoning on Words and Trees with Data*. PhD thesis, Laboratoire Spécification et Vérification, ENS Cachan, France, 2010.
- [Fig12] D. Figueira. Decidability of downward XPath. *ACM Transactions on Computational Logic*, 13(4):34, 2012.
- [Fig18] D. Figueira. Satisfiability of xpath on data trees. *ACM SIGLOG News*, 5(2):4–16, 2018.
- [Fin75] K. Fine. Normal forms in modal logic. *Notre Dame Journal of Formal Logic*, 16(2):229–237, 1975.
- [FS11] D. Figueira and L. Segoufin. Bottom-up automata on data trees and vertical XPath. In *28th International Symposium on Theoretical Aspects of Computer Science (STACS 2011)*, pages 93–104, 2011.
- [GKP05] G. Gottlob, C. Koch, and R. Pichler. Efficient algorithms for processing XPath queries. *ACM Transactions on Database Systems*, 30(2):444–491, 2005.
- [Gol84] R. Goldblatt. An abstract setting for Henkin proofs. *Topoi*, 3(1):37–41, Jun 1984.
- [HKT00] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, Cambridge, MA, 2000.
- [KRV15] E. Kostylev, J. Reutter, and D. Vrgoč. XPath for DL ontologies. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI’15*, pages 1525–1531. AAAI Press, 2015.
- [LMV16] L. Libkin, W. Martens, and D. Vrgoč. Querying graphs with data. *J. ACM*, 63(2):14:1–14:53, 2016.
- [LV12] L. Libkin and D. Vrgoč. Regular path queries on graphs with data. In *International Conference on Database Theory*, pages 74–85, 2012.
- [Mar04] M. Marx. XPath with conditional axis relations. In *International Conference on Extending Database Technology (EDBT’04)*, volume 2992 of *LNCIS*, pages 477–494. Springer, 2004.
- [Mdr05] M. Marx and M. de Rijke. Semantic characterizations of navigational XPath. *ACM SIGMOD Record*, 34(2):41–46, 2005.
- [MN07] W. Martens and F. Neven. Frontiers of tractability for typechecking simple xml transformations. *Journal of Computer and System Sciences*, 73(3):362–390, 2007.
- [Pra91] V. R. Pratt. Dynamic algebras: Examples, constructions, applications. *Studia Logica*, 50(3-4):571–605, 1991.
- [Rey01] M. Reynolds. An axiomatization of full computation tree logic. *The Journal of Symbolic Logic*, 66(3):1011–1057, 2001.
- [RWE13] I. Robinson, J. Webber, and E. Eifrem. *Graph Databases*. O’Reilly Media, Inc., 2013.
- [SP10] L. Schröder and D. Pattinson. Named models in coalgebraic hybrid logic. In J.-Y. Marion and Th. Schwentick, editors, *27th International Symposium on Theoretical Aspects of Computer Science, STACS 2010, March 4-6, 2010, Nancy, France*, volume 5 of *LIPICs*, pages 645–656. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010.
- [SV01] U. Sattler and M. Vardi. The hybrid mu-calculus. In R. Goré, A. Leitsch, and T. Nipkow, editors, *Proceedings of the International Joint Conference on Automated Reasoning*, volume 2083 of *LNAI*, pages 76–91. Springer Verlag, 2001.

- [tCLM10] B. ten Cate, T. Litak, and M. Marx. Complete axiomatizations for XPath fragments. *Journal of Applied Logic*, 8(2):153–172, 2010.
- [tCM09] B. ten Cate and M. Marx. Axiomatizing the logical core of XPath 2.0. *Theory of Computing Systems*, 44(4):561–589, 2009.
- [Wad00] P. Wadler. A formal semantics of patterns in XSLT and XPath. *Markup Languages*, 2(2):183–202, 2000.
- [Wor02] World Wide Web Consortium. XQuery 1.0 and XPath 2.0 formal semantics. W3C working draft. Technical report, 2002. <http://www.w3c.org/TR/query-algebra>.