Hilbert-style Axiomatization for Hybrid XPath with Data

Carlos Areces and Raul Fervari

FaMAF, Universidad Nacional de Córdoba, Argentina
CONICET, Argentina
{areces,fervari}@famaf.unc.edu.ar

Abstract. In this paper we introduce a sound and complete axiomatization for XPath with data constraints extended with hybrid operators. First, we define $\mathrm{HXPath}_{=}(\uparrow\downarrow)$, an extension of vertical XPath with nominals and the hybrid operator @. Then, we introduce an axiomatic system for $\mathrm{HXPath}_{=}(\uparrow\downarrow)$, and we prove it is complete with respect to the class of abstract data trees, i.e., data trees in which data values are abstracted as equivalence relations. As a corollary, we also obtain completeness with respect to the class of concrete data trees.

Keywords: XPath, modal logic, hybrid logic, data tree, axiomatization.

1 XPath as a Modal Logic with Data Tests

XPath is arguably the most widely used XML query language. Indeed, XPath is implemented in XSLT and XQuery and it is used in many specification and update languages. XPath 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 [14]. [21] adapts the definition of XPath to be used as a powerful query language over knowledge bases. Core-XPath [20] is the fragment of XPath 1.0 containing the navigational behavior 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* [8,10]. Core-XPath has been well studied from a modal point of view. For instance, its satisfiability problem is known to be decidable even in the presence of DTDs [22,6]. Moreover, it is known that it is equivalent to FO2 (first-order logic with two variables over an appropriate signature on trees) in terms of expressive power [23], and that it is strictly less expressive than PDL with converse over trees [7]. Sound and complete axiomatizations for Core-XPath have been introduced in [13,12].

However, from a database perspective, Core-XPath is not expressive enough 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 [11]. Here, we will call this logic XPath$_=$. Models of XPath$_=$ are 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 (see Figure 1 for an example). We will relax the condition on finiteness and consider also infinite data trees, although all our results hold also on finite structures. 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 tree using axes: descendant, child, ancestor, next-sibling, etc. and can make tests in intermediate nodes. The formula $\langle \alpha = \beta \rangle$ (respectively $\langle \alpha \neq \beta \rangle$) is true at a node x of a data tree if there are nodes y, z that can be reached by paths denoted by α, β respectively, and such that the data value of y is equal (respectively different) to the data value of z. For instance, in Figure 1 the expression *"there is a one-step descendant and a two-steps descendant sharing the same data value"* is satisfied at x, given the presence of u and z. The expression *"there are two children with distinct data value"* is also true at x, because y and z have different data.

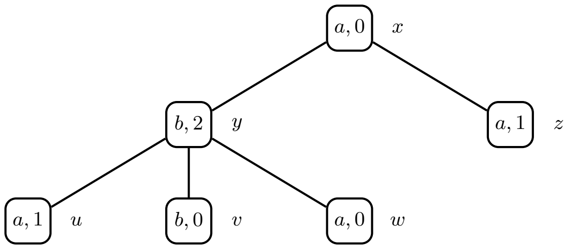

Fig. 1. An example of a data tree.

Notice that XPath$_=$ allows to compare data values at the end of a path, by equality or inequality. However, it does not allow the access to the concrete data value of nodes (in the example, 0, 1 or 2). Hence, it is possible to work with an abstraction of data trees: instead of having concrete data values in each node, we have an equivalence relation between nodes. In the data tree from Figure 1, the relation consists of three equivalence classes: $\{x, v, w\}$, $\{u, z\}$ and $\{y\}$.

Recent articles investigate XPath$_=$ from a modal perspective. For example, satisfiability and evaluation are discussed in [15,19,16], while model theory and expressivity are studied in [3,17,18,2]. We will focus in the proof theory of XPath$_=$ extended with hybrid operators. In [5], a Gentzen-style sequent calculus is given for a very restricted fragment of 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 [12] is introduced in [1], allowing downward navigation and equality/inequality tests.

In this article we will continue the investigation of axiomatic systems for XPath$_=$. In particular, we will introduce a Hilbert-style axiomatization for the logic with downward and upward navigation, 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 Vertical XPath* (denoted HXPath$_=(\uparrow\downarrow)$). We will take advantage of hybrid operators to prove completeness using a Henkin-style model construction (see [8] for details).

The article is organized as follows. In §2 we introduce the syntax and semantics of HXPath$_=(\uparrow\downarrow)$. Then we define the axiomatic system HXP in §3 and we prove its completeness in §4. In §5 we extend HXP to prove completeness with respect to the class of data trees. To conclude, in §6 we introduce some remarks and future lines of research.

2 Preliminaries

In this section we introduce the syntax and semantics for the logic we call Hybrid Vertical XPath (HXPath$_=(\uparrow\downarrow)$ for short). We assume basic knowledge of classical modal logic (see [8] for further details).

We start by defining the structures that will be used to evaluate formulas in the language.

Definition 1 (Hybrid Data Models). *Let* LAB *(the set of labels) and* NOM *(the set of nominals) be two infinite countable sets. An* abstract hybrid data model *is a tuple* $\mathcal{M} = \langle M, \sim, \to, label, nom \rangle$*, where* M *is a non-empty set of elements,* $\sim\ \subseteq M \times M$ *is an equivalence relation between elements of* M*,* $\to\ \subseteq M \times M$ *is the accessibility relation,* $label : M \to 2^{\mathsf{LAB}}$ *is a labeling function and* $nom : \mathsf{NOM} \to M$ *is a function that assigns nominals to certain elements.*

A concrete hybrid data model *is a tuple* $\mathcal{M} = \langle M, D, \to, label, nom, data \rangle$*, where* M *is a non-empty set of elements,* D *is a non-empty set of data,* $\to\ \subseteq M \times M$ *is the accessibility relation,* $label : M \to 2^{\mathsf{LAB}}$ *is the labeling function,* $nom : \mathsf{NOM} \to M$ *is a function which names the nodes and* $data : M \to D$ *is the function which assigns a data value to each node of the model.*

We often write $w{\downarrow}v$ *and* $v{\uparrow}w$ *when* $w \to v$*.*

Concrete data models are most commonly used in application, where we encounter data from an infinite alphabet (e.g., alphabetic strings) associated to the nodes in a semi-structured database. It is easy to see that each concrete data model has an associated, equivalent abstract data model where data is replaced by an equivalence relation that links all nodes with the same data. Vice-versa, each abstract data model can be "concretized" by assigning to each node its equivalence data class as data. We will prove sound and completeness over the class of abstract data models and, as a corollary, obtain completeness over concrete data models.

We are now ready to introduce the syntax and semantics of HXPath$_=(\uparrow\downarrow)$.

Definition 2 (Syntax). *The set of path expressions (which we will note as α, β, γ, ...) and node expressions (which we will note as φ, ψ, θ, ...) of HXPath$_=(\uparrow\downarrow)$ are defined by mutual recursion as follows:*

$$\begin{aligned}\alpha,\beta &::= \downarrow \mid \uparrow \mid @_i \mid [\varphi] \mid \alpha\beta \\ \varphi,\psi &::= a \mid i \mid \neg\varphi \mid \varphi\wedge\psi \mid \langle\alpha=\beta\rangle \mid \langle\alpha\neq\beta\rangle, \qquad a\in\mathsf{LAB}, i\in\mathsf{NOM}.\end{aligned}$$

Notice that path expressions occur in node expressions in *data comparison formulas* of the form $\langle\alpha=\beta\rangle$ and $\langle\alpha\neq\beta\rangle$, while node expressions occur in path expressions in *test formulas* of the form $[\varphi]$.

In what follows we will always use δ to represent the $\downarrow$ and $\uparrow$ operators and $*$ for $=$ and $\neq$. Other Boolean operators are defined as usual. We define the following operators as abbreviations.

Definition 3 (Abbreviations). *Let α,β 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* LAB*:*

Node Expressions	**Path Expressions**
$\top \equiv p\vee\neg p$	$\epsilon \equiv [\top]$
$\bot \equiv \neg\top$	$\langle\gamma_1(\alpha\cup\beta)\gamma_2 * \gamma_3\rangle \equiv \langle\gamma_1\alpha\gamma_2 * \gamma_3\rangle \vee \langle\gamma_1\beta\gamma_2 * \gamma_3\rangle$
$\langle\alpha\rangle\varphi \equiv \langle\alpha[\varphi]=\alpha[\varphi]\rangle$	$\langle\gamma_1 * \gamma_2(\alpha\cup\beta)\gamma_3\rangle \equiv \langle\gamma_1 * \gamma_2\alpha\gamma_3\rangle \vee \langle\gamma_1 * \gamma_2\beta\gamma_3\rangle$
$[\alpha]\varphi \equiv \neg\langle\alpha\rangle\neg\varphi$	
$@_i\varphi \equiv \langle @_i\rangle\varphi$	

As a corollary of the definition 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 "at" 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 by 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 $[\varphi]\alpha$ 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.

Definition 4 (Semantics). *Let $\mathcal{M}=\langle M,\sim,\rightarrow,label,nom\rangle$ be an abstract data model, and $x,y\in M$. We define the semantics of HXPath$_=(\uparrow\downarrow)$ as follows:*

$$\begin{aligned}
\mathcal{M},x,y \models \downarrow \quad &\textit{iff} \quad x\rightarrow y \\
\mathcal{M},x,y \models \uparrow \quad &\textit{iff} \quad y\rightarrow x \\
\mathcal{M},x,y \models @_i \quad &\textit{iff} \quad nom(i)=y \\
\mathcal{M},x,y \models [\varphi] \quad &\textit{iff} \quad x=y \textit{ and } \mathcal{M},x\models\varphi \\
\mathcal{M},x,y \models \alpha\beta \quad &\textit{iff} \quad \textit{there is some } z\in M \textit{ s.t. } \mathcal{M},x,z\models\alpha \textit{ and } \mathcal{M},z,y\models\beta \\
\mathcal{M},x \models a \quad &\textit{iff} \quad a\in label(x) \\
\mathcal{M},x \models i \quad &\textit{iff} \quad nom(i)=x \\
\mathcal{M},x \models \neg\varphi \quad &\textit{iff} \quad \mathcal{M},x\not\models\varphi \\
\mathcal{M},x \models \varphi\wedge\psi \quad &\textit{iff} \quad \mathcal{M},x\models\varphi \textit{ and } \mathcal{M},x\models\psi \\
\mathcal{M},x \models \langle\alpha=\beta\rangle \quad &\textit{iff} \quad \textit{there are } y,z{\in}M \textit{ s.t. } \mathcal{M},x,y\models\alpha,\ \mathcal{M},x,z\models\beta \textit{ and } y\sim z \\
\mathcal{M},x \models \langle\alpha\neq\beta\rangle \quad &\textit{iff} \quad \textit{there are } y,z{\in}M \textit{ s.t. } \mathcal{M},x,y\models\alpha,\ \mathcal{M},x,z\models\beta \textit{ and } y\not\sim z.
\end{aligned}$$

Corollary 1.

$$\begin{array}{rll} \mathcal{M}, x \models @_i\varphi & \textit{iff} & \mathcal{M}, nom(i) \models \varphi \\ \mathcal{M}, x \models \langle\delta\rangle\varphi & \textit{iff} & \textit{there is some } y \in M \textit{ s.t. } x\delta y \textit{ and } \mathcal{M}, y \models \varphi \\ \mathcal{M}, x \models [\delta]\varphi & \textit{iff} & \textit{for all } y \in M,\ x\delta y \textit{ then } \mathcal{M}, y \models \varphi. \end{array}$$

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

Example 1. We list below some HXPath$_=(\uparrow\downarrow)$ expressions together with their intuitive meaning:

$\alpha[i]$	There exists an α path between the current point of evaluation and the node named i.
$@_i\alpha$	There exists an α path between the node named i and some other node.
$\langle @_i = @_j \rangle$	The node named i has the same data than the node named j.
$\langle \alpha = @_i\beta \rangle$	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.

3 Axiomatic System

In this section we introduce the axiomatic system HXP for HXPath$_=(\uparrow\downarrow)$. It is an extension of an axiomatic system for the hybrid logic HL(@) which adds nominals and the @ operator to the basic modal language (see [8]). In particular, we include axioms to handle data equality and inequality.

We present axioms and rules step by step, providing brief comments to help the reader understand their role. In all cases, we provide *axiom and rule schemes*, i.e., they can be instantiated with arbitrary path and node expressions (but always respecting types). In all axioms and rules φ, ψ and θ are node expressions, α, β and γ are path expressions, i, j and k are nominals. We use $\vdash \varphi$ to indicate that φ is a theorem of HXP.

In addition to an arbitrary set of axiom and rule schemes for propositional logic, we include 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 $[\alpha](\varphi \to \psi) \to ([\alpha]\varphi \to [\alpha]\psi)$	$\dfrac{\vdash \varphi}{\vdash [\alpha]\varphi}\,Nec$

Then we introduce generalizations of the rules for the hybrid logic HL(@).

Hybrid rules

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

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

Now we introduce axioms that handle @. Notice that $@_i$ is a path expression of HXPath$_=(\uparrow\downarrow)$ 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	$\neg @_i\varphi \leftrightarrow @_i\neg\varphi$	@-refl.	$@_i i$
@-intro	$i \wedge \varphi \to @_i\varphi$	@-sym.	$@_i j \to @_j i$
		nom	$@_i j \wedge \langle @_i\alpha * \beta\rangle \to \langle @_j\alpha * \beta\rangle$
		agree	$\langle @_j @_i\alpha * \beta\rangle \leftrightarrow \langle @_i\alpha * \beta\rangle$
		back	$\langle \gamma @_i\alpha * \beta\rangle \to \langle @_i\alpha * \beta\rangle$

Axioms involving the classical XPath operators can be found below. We organize them in three groups. First, we have axioms for the interaction between $\downarrow$ and $\uparrow$. These axioms are the classical ones characterizing “future” and “past” modalities (see [8]). Then, we introduce axioms to handle complex path expressions in data comparisons. Finally, we introduce axioms to handle data tests.

Axioms for $\downarrow, \uparrow$-interaction	
down-up	$\varphi \to [\downarrow]\langle\uparrow\rangle\varphi$
up-down	$\varphi \to [\uparrow]\langle\downarrow\rangle\varphi$
Axioms for paths	
comp-assoc	$\langle(\alpha\beta)\gamma * \eta\rangle \leftrightarrow \langle\alpha(\beta\gamma) * \eta\rangle$
comp-neutral	$\langle\alpha\beta * \gamma\rangle \leftrightarrow \langle\alpha\epsilon\beta * \gamma\rangle$ (α or β can be empty)
comp-dist	$\langle\alpha\beta\rangle\varphi \leftrightarrow \langle\alpha\rangle\langle\beta\rangle\varphi$
Axioms for data	
equal	$\langle\epsilon = \epsilon\rangle$
distinct	$\neg\langle\epsilon \neq \epsilon\rangle$
@-data	$\neg\langle @_i{=}@_j\rangle \leftrightarrow \langle @_i{\neq}@_j\rangle$
ϵ-trans	$\langle\epsilon = \alpha\rangle \wedge \langle\epsilon = \beta\rangle \to \langle\alpha = \beta\rangle$
$*$-comm	$\langle\alpha * \beta\rangle \leftrightarrow \langle\beta * \alpha\rangle$
$*$-test	$\langle[\varphi]\alpha * \beta\rangle \leftrightarrow \varphi \wedge \langle\alpha * \beta\rangle$
$@*$-dist	$\langle @_i\alpha * @_i\beta\rangle \leftrightarrow @_i\langle\alpha * \beta\rangle$
comp$*$-dist	$\langle\alpha\rangle\langle\beta * \gamma\rangle \to \langle\alpha\beta * \alpha\gamma\rangle$

Proposition 1. *The following formulas are theorems in* HXP.

1. *test-dist* $\quad \vdash \langle[\varphi] = [\psi]\rangle \leftrightarrow \varphi \wedge \psi$

2. test-$\bot$ $\vdash \langle [\varphi] \neq [\psi] \rangle \leftrightarrow \langle \epsilon \neq \epsilon \rangle$
3. @*-swap* $\vdash @_i \langle \alpha * @_j \beta \rangle \leftrightarrow @_j \langle \beta * @_i \alpha \rangle$
4. bridge $\vdash \langle \alpha \rangle i \wedge @_i \varphi \to \langle \alpha \rangle \varphi$

Proof. (test-dist and test-$\bot$*).* Let $*$ be $=$ or $\neq$. Then:
$\vdash \langle [\varphi] * [\psi] \rangle \leftrightarrow \langle [\varphi]\epsilon * [\psi] \rangle$ by *comp-neutral.*
$\vdash \langle [\varphi]\epsilon * [\psi] \rangle \leftrightarrow \varphi \wedge \langle \epsilon * [\psi] \rangle$ by $*$*-test.*
$\vdash \varphi \wedge \langle \epsilon * [\psi] \rangle \leftrightarrow \varphi \wedge \langle [\psi] * \epsilon \rangle$ by $*$*-comm.*
$\vdash \varphi \wedge \langle [\psi] * \epsilon \rangle \leftrightarrow \varphi \wedge \langle [\psi]\epsilon * \epsilon \rangle$ by *comp-neutral.*
$\vdash \varphi \wedge \langle [\psi]\epsilon * \epsilon \rangle \leftrightarrow \varphi \wedge \psi \wedge \langle \epsilon * \epsilon \rangle$ by $*$*-test.*
Replacing $*$ by $=$ we get $\varphi \wedge \psi$ by *equal.* Replacing it by $\neq$ we get $\langle \epsilon \neq \epsilon \rangle$.

(@*-swap).* $\vdash @_i \langle \alpha = @_j \beta \rangle \leftrightarrow \langle @_i \alpha = @_i @_j \beta \rangle$ by @ $=$*-dist.*
$\vdash \langle @_i \alpha = @_i @_j \beta \rangle \leftrightarrow \langle @_i @_j \beta = @_i \alpha \rangle$ by $=$*-comm.*
$\vdash \langle @_i @_j \beta = @_i \alpha \rangle \leftrightarrow \langle @_j \beta = @_i \alpha \rangle$ by *agree.*
$\vdash \langle @_j \beta = @_i \alpha \rangle \leftrightarrow \langle @_i \alpha = @_j \beta \rangle$ by $=$*-comm.*
$\vdash \langle @_i \alpha = @_j \beta \rangle \leftrightarrow \langle @_j @_i \alpha = @_j \beta \rangle$ by *agree.*
$\vdash \langle @_j @_i \alpha = @_j \beta \rangle \leftrightarrow @_j \langle @_i \alpha = \beta \rangle$ by *agree.*
$\vdash @_j \langle @_i \alpha = \beta \rangle \leftrightarrow @_j \langle \beta = @_i \alpha \rangle$ by $=$*-comm.*

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

4 Completeness

It is a fairly straightforward exercise to prove that the axioms and rules of HXP are sound for the intended semantics. We will now show that the axiomatic system is also complete. The completeness argument follows the lines of the completeness proof for HL(@) (see [8]), which is a Henkin-style proof with nominals playing the role of first-order constants.

In what follows, we will write $\Gamma \vdash \varphi$ if and only if φ can be obtained from a set of formulas Γ by applying the inference rules of HXP.

Definition 5. *Let Γ be a set of formulas, we say that Γ is an* HXP maximal consistent set *(*HXP*-MCS, or MCS for short) if and only if $\Gamma \nvdash \bot$ and for all $\varphi \notin \Gamma$ we have $\Gamma \cup \{\varphi\} \vdash \bot$.*

Proposition 2. *Let Γ be an* HXP*-MCS. Then, the following facts hold:*

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

Proof. Item 1 is a consequence of @*-intro*, 2 follows from @$=$*-dist* and 3 can be proved using *agree* and $=$*-comm.* □

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

Corollary 2. *Let Γ be a MCS. Then for all φ, either $\varphi \in \Gamma$ or $\varphi \notin \Gamma$.*

In the same way as for hybrid logic, inside every MCS there are a collection of MCSs with some desirable properties:

Lemma 1. *Let Γ be an HXP-MCS. For any nominal $i \in \Gamma$, let us define $\Delta_i = \{\varphi \mid @_i\varphi \in \Gamma\}$. Then*

1. *Δ_i is an HXP-MCS.*
2. *For all nominals i, j, if $i \in \Delta_j$ then $\Delta_i = \Delta_j$.*
3. *For all nominals i, j, we have $@_i\varphi \in \Delta_j$ iff $@_i\varphi \in \Gamma$.*
4. *If $k \in \Gamma$ then $\Gamma = \Delta_k$.*

Proof. See [8, Lemma 7.24] for details.

Definition 6 (Named and Pasted MCS). *Let Γ be an HXP-MCS. We say that Γ is* named *if for some nominal i we have that $i \in \Gamma$ (and we will say that Γ is named by i). We say that Γ is* pasted *if the following holds:*

1. *$\langle @_i\delta\alpha = \beta\rangle \in \Gamma$ implies that for some nominal j, $@_i\langle\delta\rangle j \wedge \langle @_j\alpha = \beta\rangle \in \Gamma$*
2. *$\langle @_i\delta\alpha \neq \beta\rangle \in \Gamma$ implies that for some nominal j, $@_i\langle\delta\rangle j \wedge \langle @_j\alpha \neq \beta\rangle \in \Gamma$.*

Now we are going to prove a crucial property in our completeness proof: the *Extended Lindenbaum Lemma.* Intuitively, it says that the rules of HXP 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 2 (Extended Lindenbaum Lemma). *Let NOM' be a (countably) infinite set of nominals disjoint from NOM, and let $HXPath_=(\uparrow\downarrow)'$ be the language obtained by adding these new nominals to $HXPath_=(\uparrow\downarrow)$. Then, every HXP-consistent set of formulas in $HXPath_=(\uparrow\downarrow)$ can be extended to a named and pasted HXP-MCS in $HXPath_=(\uparrow\downarrow)'$.*

Proof. Enumerate NOM'. Given Σ a consistent set in $\mathrm{HXPath}_=(\uparrow\downarrow)$, define Σ_k to be $\Sigma \cup \{k\}$, where k is the first nominal in our enumeration. Σ_k is consistent, otherwise for some conjunction θ from Σ, $\vdash k \to \neg\theta$. By the *name* rule, $\vdash \neg\theta$, contradicting the consistency of Σ.

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

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

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

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

Definition 7 (Extracted Model). *Let Γ be a named and pasted* HXP*-MCS, then we define the* extracted model *from Γ, $\mathcal{M}_\Gamma = \langle M, \sim, \rightarrow, \mathit{label}, \mathit{nom}\rangle$ as:*

- $M = \{\Delta_i \mid \ \Delta_i$ *was obtained from* $\Gamma\}$
- $\Delta_i \rightarrow \Delta_j$ *iff* $\langle\downarrow\rangle j \in \Delta_i$
- $a \in \mathit{label}(\Delta_i)$ *iff* $a \in \Delta_i$
- $\mathit{nom}(i) = \Delta_i$
- $\Delta_i \sim \Delta_j$ *iff* $\langle \epsilon = @_j\rangle \in \Delta_i$.

Proposition 3. *Let $\mathcal{M}_\Gamma = \langle M, \sim, \rightarrow, \mathit{label}, \mathit{nom}\rangle$ be the extracted model, for some Γ. Then,*

1. *$\Delta_i \rightarrow \Delta_j$ if and only if $\langle\uparrow\rangle i \in \Delta_j$, and*
2. *$\Delta_i \not\sim \Delta_j$ if and only if $\langle \epsilon \neq @_j\rangle \in \Delta_i$.*

Proof. Item 1 uses the same argument as for HL(@) in addition to the axioms for ↑; item 2 follows from @-*data*.

We need to prove that, in fact, $\mathcal{M}_\Gamma$ is an abstract hybrid data model.

Proposition 4. *$\mathcal{M}_\Gamma$ is well defined, i.e., the following properties hold:*

1. *$\mathit{nom}(i) = \Delta_1$ and $\mathit{nom}(i) = \Delta_2$ then $\Delta_1 = \Delta_2$, and*
2. *$\sim$ is an equivalence relation.*

Proof. Item 1 follows from the axioms for the hybrid operators in a standard way. Let us prove that $\sim$ is an equivalence relation.
- *Reflexivity:* $\Delta_i \sim \Delta_i$ iff $\langle \epsilon = @_i\rangle \in \Delta_i$ iff $@_i\langle \epsilon = \epsilon\rangle \in \Delta_i$, which is true because $\langle \epsilon = \epsilon\rangle$ is a theorem.
- *Symmetry:* $\Delta_i \sim \Delta_j$ iff $\langle \epsilon = @_j\rangle \in \Delta_i$. By definition of Δ_i, we have $@_i\langle \epsilon = @_j\rangle \in \Gamma$, and by *neutral* and =-*comm* we get $@_i\langle \epsilon = @_j \epsilon\rangle \in \Gamma$. Then, by @-*swap* $@_j\langle \epsilon = @_i \epsilon\rangle$. Therefore $\langle \epsilon = @_i\rangle \in \Delta_j$ (by neutral), iff $\Delta_j \sim \Delta_i$.
- *Transitivity:* Suppose $\Delta_i \sim \Delta_j$ and $\Delta_j \sim \Delta_k$, iff $\langle \epsilon = @_j\rangle \in \Delta_i$ and $\langle \epsilon = @_k\rangle \in \Delta_j$. This means that we have $@_i\langle \epsilon = @_j\rangle \in \Gamma$ iff (by @-swap) $@_j\langle \epsilon = @_i\rangle \in \Gamma$, and $@_j\langle \epsilon = @_k\rangle \in \Gamma$. Then $\langle \epsilon = @_i\rangle \wedge \langle \epsilon = @_k\rangle \in \Delta_j$, and by ϵ-*trans* we have $\langle @_i = @_k\rangle \in \Delta_j$. By *agree* and @=-*dist* we get $@_i\langle \epsilon = @_k\rangle \in \Delta_j$, iff by definition of Δ_j, $@_j @_i\langle \epsilon = @_k\rangle \in \Gamma$. By *agree* we obtain $@_i\langle \epsilon = @_k\rangle \in \Gamma$, then $\langle \epsilon = @_k\rangle \in \Delta_i$. Hence, we have $\Delta_i \sim \Delta_k$. □

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

Lemma 3 (Existence Lemma). *Let Γ be an* HXP*-MCS and let $\mathcal{M}_\Gamma = \langle M, \sim, \rightarrow, \mathit{label}, \mathit{nom}\rangle$ be the extracted model from Γ. Suppose $\Delta \in M$ and $i \in \Delta$. Then*

1. *$\langle \delta\alpha = \beta \rangle \in \Delta$ implies there exists $\Sigma \in M$ s.t. $\Delta\delta\Sigma$ and $\langle \alpha = @_i\beta \rangle \in \Sigma$,*
2. *$\langle \delta\alpha \neq \beta \rangle \in \Delta$ implies there exists $\Sigma \in M$ s.t. $\Delta\delta\Sigma$ and $\langle \alpha \neq @_i\beta \rangle \in \Sigma$.*

Proof. We discuss the case for $=$ (the case for $\neq$ is similar). Because $\Delta \in M$, for some nominal i we have $\Delta = \Delta_i$. As $\langle \delta\alpha = \beta \rangle \in \Delta$, $@_i\langle \delta\alpha = \beta \rangle \in \Gamma$. Then, by Axiom $@{=}$-*dist*, $\langle @_i\delta\alpha = @_i\beta \rangle \in \Gamma$. Because Γ is pasted $@_i\langle \delta \rangle j \wedge \langle @_j\alpha = @_i\beta \rangle \in \Gamma$. As Γ is MCS, $@_i\langle \delta \rangle j \in \Gamma$ and $\langle @_j\alpha = @_i\beta \rangle \in \Gamma$. By Axiom *agree*, we have $\langle @_j\alpha = @_j@_i\beta \rangle \in \Gamma$. Then, $@_j\langle \alpha = @_i\beta \rangle \in \Gamma$ by $@{=}$-*dist*. By definition, $\langle \delta \rangle j \in \Delta_i$ and $\langle \alpha = @_i\beta \rangle \in \Delta_j$. Taking Σ as Δ_j, we complete the proof. □

Now we are ready to prove the Truth Lemma that states that membership in an MCS of the extracted model is equivalent to being true in that MCS.

Lemma 4 (Truth Lemma). *Let $\mathcal{M}_\Gamma = \langle M, \sim, \rightarrow, label, nom \rangle$ be the extracted model from a MCS Γ, and let $\Delta_i \in M$. Then, for any formula φ,*

$$\mathcal{M}_\Gamma, \Delta_i \models \varphi \quad \textit{iff} \quad \varphi \in \Delta_i.$$

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

The proof proceeds by induction in the complexity of φ and α. First, we prove the base cases:

- $\alpha = \downarrow$: Suppose $\mathcal{M}_\Gamma, \Delta_i, \Delta_j \models \downarrow$ iff $\Delta_i \rightarrow \Delta_j$ (by $\models$), iff $\langle \downarrow \rangle j \in \Delta_i$ (by definition of extracted model).

- $\alpha = \uparrow$: Suppose $\mathcal{M}_\Gamma, \Delta_i, \Delta_j \models \uparrow$ iff $\Delta_j \rightarrow \Delta_i$ (by $\models$), iff $\langle \uparrow \rangle j \in \Delta_i$ (by 1 of Proposition 3).

- $\alpha = @_k$: 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 $j \in \Delta_j$ we have $j \in \Delta_k$. Then, we have $@_k j \in \Gamma$, and by Axiom *agree*, $@_i@_k j \in \Gamma$. Therefore, $@_k j \in \Delta_i$.

- $\varphi = a$: $\mathcal{M}_\Gamma, \Delta_i \models a$ iff $a \in label(\Delta_i)$, iff $a \in \Delta_i$.

- $\varphi = j$: $\mathcal{M}_\Gamma, \Delta_i \models j$ iff $nom(j) = \Delta_i$, iff $\Delta_i = \Delta_j$ iff $j \in \Delta_i$.

Now we prove the inductive cases:

- $\varphi = \psi \wedge \rho$ 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 $\psi \in \Delta_i$ and $j \in \Delta_i$. By Δ_i MCS, we have $\psi \wedge j \in \Delta_i$, and by idempotence of the conjunction we have $\psi \wedge \psi \wedge j \wedge j \in \Delta_i$. Also, we have $\langle \epsilon = \epsilon \rangle \in \Delta_i$, then we can use Axioms $=$-*test* and $=$-*comm* to obtain $\langle [\psi][j] = [\psi][j] \rangle \in \Delta_i$ (which is the same as $\langle [\psi] \rangle j$) as we wanted.

- $\alpha = \beta\gamma$: $\mathcal{M}_\Gamma, \Delta_i, \Delta_j \models \beta\gamma$ iff there is some Δ_k such that $\mathcal{M}_\Gamma, \Delta_i, \Delta_k \models \beta$ and $\mathcal{M}_\Gamma, \Delta_k, \Delta_j, \models \gamma$. By (IH2), we have $\langle \beta \rangle k \in \Delta_i$ and $\langle \gamma \rangle j \in \Delta_k$. We can conclude $@_i\langle \beta \rangle k \in \Gamma$ and $@_k\langle \gamma \rangle j \in \Gamma$, then $@_i\langle \beta \rangle k \wedge @_k\langle \gamma \rangle j \in \Gamma$. By *agree*, we have $@_i\langle\beta\rangle k \wedge @_i@_k\langle\gamma\rangle j \in \Gamma$, and with a very simple hybrid argument we get $@_i(\langle\beta\rangle k \wedge @_k\langle\gamma\rangle j) \in \Gamma$. By *bridge*, we have $@_i(\langle\beta\rangle\langle\gamma\rangle j) \in \Gamma$, and by Axiom *comp-dist* $@_i(\langle\beta\gamma\rangle j) \in \Gamma$. Hence, $\langle\beta\gamma\rangle j \in \Delta_i$.

For node expressions of the form $\langle\alpha * \beta\rangle$ we need to do induction on the length of α and β (defined in the obvious way).

First notice that by $*$-*comm*, $\langle\alpha * \beta\rangle \in \Delta_i$ iff $\langle\beta * \alpha\rangle \in \Delta_i$. 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 discuss the case 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.

- $\varphi = \langle[\psi] = [\rho]\rangle$: direct from *test-dist*.

- $\varphi = \langle[\psi] \neq [\rho]\rangle$: it is a contradiction from *test-*$\bot$, then this case has not to be considered.

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

- $\varphi = \langle{\downarrow}\beta = \gamma\rangle$: $\mathcal{M}_\Gamma, \Delta_i \models \langle{\downarrow}\beta = \gamma\rangle$ iff there are Δ_j, Δ_k such that $\mathcal{M}_\Gamma, \Delta_i, \Delta_j \models {\downarrow}\beta$, $\mathcal{M}_\Gamma, \Delta_i, \Delta_k \models \gamma$ and $\Delta_j \sim \Delta_k$. Then, by (IH2) and definition of $\mathcal{M}_\Gamma$ we have:

1. $\langle{\downarrow}\beta\rangle j \in \Delta_i$,
2. $\langle\gamma\rangle k \in \Delta_i$, and
3. $\langle\epsilon = @_k\rangle \in \Delta_j$.

By 1 and the Existence Lemma, there is Δ_l such that both $\Delta_i \to \Delta_j$ and $\mathcal{M}_\Gamma, \Delta_l, \Delta_j \models \beta$ hold. Then by (IH2) we have

4. $\langle\beta\rangle j \in \Delta_l$.

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

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

Then, because $\Delta_i \to \Delta_l$ and the previous paragraph, we can use the Existence Lemma to obtain $\langle{\downarrow}\beta = @_i\gamma\rangle \in \Delta_i$, if and only if $\langle{\downarrow}\beta = \gamma\rangle \in \Delta_i$ (by @=-dist), hence $\langle{\downarrow}\beta = \gamma\rangle \in \Delta_i$, as we wanted.

- $\varphi = \langle{\uparrow}\beta = \gamma\rangle$ and $\varphi = \langle\delta\beta * \gamma\rangle$ are similar to the previous one but using also Proposition 3.

- $\varphi = \langle @_j\beta = \gamma\rangle$: $\mathcal{M}_\Gamma, \Delta_i \models \langle @_j\beta = \gamma\rangle$ 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 $\Delta_k \sim \Delta_l$. Then, by (IH2) and definition of $\mathcal{M}_\Gamma$ we have:

1. $\langle @_j\beta\rangle k \in \Delta_i$, iff $@_i\langle @_j\beta\rangle k \in \Gamma$ iff $@_j\langle\beta\rangle k \in \Gamma$,
2. $\langle\gamma\rangle l \in \Delta_i$, iff $@_i\langle\beta\rangle l \in \Gamma$, and

3. $\langle \epsilon = @_k \rangle \in \Delta_j$, iff $@_k\langle \epsilon = @_l \rangle \in \Gamma$.

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

- $\varphi = \langle[\psi]\beta = \gamma\rangle$: $\mathcal{M}_\Gamma, \Delta_i \models \langle[\psi]\beta = \gamma\rangle$ 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 $\Delta_j \sim \Delta_k$. Then, by (IH2) and definition of $\mathcal{M}_\Gamma$ we have:

1. $\langle\beta\rangle j \in \Delta_i$,
2. $\langle\gamma\rangle k \in \Delta_i$,
3. $\langle\epsilon = @_k\rangle \in \Delta_j$, and
4. $\psi \in \Delta_i$.

Using the same argument as in $(\otimes)$ the proof that $\langle[\psi]\beta = \gamma\rangle \in \Delta_i$ is straightforward.

- Cases involving $\neq$ are analogous, using Proposition 3 to obtain $\langle\epsilon = @_k\rangle \notin \Delta_j$ in item 3 above. □

As a result we obtain the completeness result.

Theorem 1. *The axiomatic system* HXP *is complete for abstract hybrid data models.*

Proof. We need to prove that every HXP-consistent set of HXPath$_=(\uparrow\downarrow)$-formulas Σ it satisfiable in a countable hybrid model. For any Σ, we can use the Extended Lindenbaum Lemma to obtain Σ^+ which is named and pasted in HXPath$_=(\uparrow\downarrow)'$. Let $\mathcal{M} = \langle M, \sim, \rightarrow, \mathit{label}, \mathit{nom}\rangle$ be the extracted model from Σ^+. As Σ^+ is named, then $\Sigma^+ \in M$. Then by Truth Lemma, for all $\varphi \in \Sigma$ we have $\mathcal{M}, \Sigma^+ \models \varphi$. Because each state is named by some nominal from a countable set NOM', the model is countable.

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

Corollary 3. *The axiomatic system* HXP *is complete for concrete hybrid data models.*

5 Completeness for Tree Models

As we mentioned in the introductory section, XPath$_=$ is a query language for XML documents, and that it is possible to work with some abstractions called *data trees*. So far, we introduced an axiomatic system which is sound and complete with respect of a more general class of structures, which are the hybrid data models from Definition 1. We will show that it is possible to extend the axiomatic system HXP to handle data trees, the most interesting structures for HXPath$_=(\uparrow\downarrow)$ applications.

The table below introduces two groups of axioms. Those in the first column guarantee that the evaluation model is a tree. In the second column, we have two axioms which impose a standard property required in abstractions of XML documents: the set of labels LAB is assumed to be finite and each node is labeled exactly by one tag name.

Axioms for trees		**Axioms for labels**	
no-circle	$i \to \neg\langle\downarrow\rangle^n i, n \geq 1$	lab-some	$\bigvee_{a\in\mathsf{LAB}} a$
no join	$\langle\uparrow\rangle i \wedge \langle\uparrow\rangle j \to @_i j$	lab-uniq	$\neg(a \wedge b)$ (for $a \neq b$)

We need to consider a point-generated sub-model of $\mathcal{M}_\Gamma$ to ensure that the resulting model is a tree.

Definition 8 (Generated Sub-model). *Let Γ be a named and pasted MCS using the axiomatic system* HXP *extended with the axioms for trees and labels, and $\mathcal{M}_\Gamma = \langle M, \sim, \to, label, nom\rangle$ the extracted model from Γ. We define $\mathcal{T}_\Gamma$ as the point-generated sub-model of $\mathcal{M}_\Gamma$ obtained from Γ, i.e., $\mathcal{T}_\Gamma$ is the smallest sub-model of $\mathcal{M}_\Gamma$ that includes Γ in its domain, and such that for all points w, the following closure condition holds:*

$$\text{If } w \in \mathcal{T}_\Gamma \text{ and } w \to v, \text{ then } v \in \mathcal{T}_\Gamma.$$

Proposition 5. *$\mathcal{T}_\Gamma$ is a tree.*

Proof. By construction Γ is the root of $\mathcal{T}_\Gamma$. We have to prove that the accessibility relation is a) irreflexive, b) asymmetric and c) that every node except the root has exactly one immediate predecessor. The proof is standard using axioms for $\downarrow, \uparrow$ interaction and the axioms for trees. □

It should be obvious that the axioms for labels ensure that exactly one label holds in a node. Using $\mathcal{T}_\Gamma$ in the Truth Lemma gives the desired result.

Theorem 2. *The axiomatic system* HXP *extended with the axioms for trees and labels is complete for abstract named data trees (and consequently, for concrete named data trees).*

6 Final Remarks

We introduced a sound and complete axiomatization for HXPath$_=(\uparrow\downarrow)$, i.e., the language XPath with upward and downward navigation and data comparisons, extended with nominals and the hybrid operator @. The *hybridization* of XPath allowed us to replicate the completeness argument for the hybrid logic HL(@) shown, e.g., in [8].

As future work we would like to take advantage of the hybridization of XPath$_=$ to obtain general axiomatizations as in [9,4]. The idea is to define minimal proof systems that are not only complete for the class of all models, but which can also be extended with additional axioms that are *pure* in some sense, ensuring completeness with respect to the corresponding class of models. Our goal is to explore this general framework and obtain complete axiomatic systems for some natural extensions of HXPath$_=(\uparrow\downarrow)$:

- HXPath$_=(\uparrow\downarrow)$ with reflexive-transitive closure for downward/upward navigation (i.e., allowing $\downarrow^*$ and $\uparrow^*$), and sibling navigation.
- Exploring new kind of data comparisons, for instance, including the relation $<$ in addition to $=$ and $\neq$.

Another aspect we would like to explore is *decidability* and *complexity*. A filtration argument (see [8]) can be applied to prove that HXPath$_=(\uparrow\downarrow)$ is decidable over the class of all models, obtaining a NExpTime upper bound for the satisfiability problem. We conjecture that the satisfiability problem is also decidable over the class of finite data trees, and that this result can be proved adapting the automata proof given in [15], with the method used to account for hybrid operators presented in [24].

Acknowledgments: This work was partially supported by grant ANPCyT-PICT-2013-2011, STIC-AmSud "Foundations of Graph Structured Data (FoG)" and the Laboratoire International Associé "INFINIS."

References

1. S. Abriola, M. Descotte, R. Fervari, and S. Figueira. Axiomatizations for downward XPath on data trees. *CoRR*, abs/1605.04271, 2016.
2. S. Abriola, M. Descotte, and S. Figueira. Model theory of XPath on data trees. Part II: Binary bisimulation and definability. Information and Computation, to appear, http://www.glyc.dc.uba.ar/santiago/papers/xpath-part2.pdf.
3. 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.
4. C. Areces and B. ten Cate. Hybrid logics. In P. Blackburn, F. Wolter, and J. van Benthem, editors, *Handbook of Modal Logics*, pages 821–868. Elsevier, 2006.
5. 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.
6. M. Benedikt, W. Fan, and F. Geerts. XPath satisfiability in the presence of DTDs. *Journal of the ACM*, 55(2):1–79, 2008.
7. M. Benedikt and C. Koch. XPath leashed. *ACM Computing Surveys*, 41(1), 2008.
8. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001.
9. P. Blackburn and B. ten Cate. Pure extensions, proof rules, and hybrid axiomatics. *Studia Logica*, 84(2):277–322, 2006.

10. P. Blackburn and J. van Benthem. Modal Logic: A Semantic Perspective. In *Handbook of Modal Logic*, pages 1–84. Elsevier, 2006.
11. 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.
12. B. ten Cate, T. Litak, and M. Marx. Complete axiomatizations for XPath fragments. *Journal of Applied Logic*, 8(2):153–172, 2010.
13. B. ten Cate and M. Marx. Axiomatizing the logical core of XPath 2.0. *Theory of Computing Systems*, 44(4):561–589, 2009.
14. J. Clark and S. DeRose. XML path language (XPath). Website, 1999. W3C Recommendation. http://www.w3.org/TR/xpath.
15. D. Figueira. *Reasoning on Words and Trees with Data*. PhD thesis, Laboratoire Spécification et Vérification, ENS Cachan, France, 2010.
16. D. Figueira. Decidability of downward XPath. *ACM Transactions on Computational Logic*, 13(4):34, 2012.
17. 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.
18. 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.
19. 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.
20. G. Gottlob, C. Koch, and R. Pichler. Efficient algorithms for processing XPath queries. *ACM Transactions on Database Systems*, 30(2):444–491, 2005.
21. 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.
22. M. Marx. XPath with conditional axis relations. In *International Conference on Extending Database Technology (EDBT'04)*, volume 2992 of *LNCS*, pages 477–494. Springer, 2004.
23. M. Marx and M. de Rijke. Semantic characterizations of navigational XPath. *ACM SIGMOD Record*, 34(2):41–46, 2005.
24. 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.