Definability for Downward and Vertical XPath on Data Trees

Sergio Abriola[1], María Emilia Descotte[1], and Santiago Figueira[1,2]

[1] University of Buenos Aires, Argentina
[2] CONICET, Argentina

Abstract. We study the expressive power of the downward and vertical fragments of XPath equipped with (in)equality tests over data trees. We give necessary and sufficient conditions for a class of pointed data trees to be definable by a set of formulas or by a single formula of each of the studied logics. To do so, we introduce a notion of saturation, and show that over saturated data trees bisimulation coincides with logical equivalence.

Keywords: XPath · data tree · bisimulation · definability · first-order logic · ultraproduct · saturation.

1 Introduction

The abstraction of an XML document is a data tree, i.e. a tree whose every node contains a tag or label (such as *LastName*) from a finite domain, and a data value (such as *Smith*) from an infinite domain. XPath is the most widely used query language for XML documents; it is an open standard and constitutes a World Wide Web Consortium (W3C) Recommendation [5]. XPath has syntactic operators to navigate the tree using the 'child', 'parent', 'sibling', etc. accessibility relations, and can make tests on intermediate nodes. Core-XPath [9] is the fragment of XPath 1.0 containing only the navigational behavior of XPath. It can express properties of the underlying tree structure of the XML document, such as "*the root of the tree has a child labeled a and a child labeled b*", but it cannot express conditions on the actual data contained in the attributes, such as "*the root of the tree has two children with same tag a but different data value*". However, Core-Data-XPath [3], here called XPath$_=$, can. Indeed, XPath$_=$ is the extension of Core-XPath with (in)equality tests between attributes of elements in an XML document.

In a recent paper [8], the expressive power of XPath$_=$ was studied, from a logical and modal model theoretical point of view. A notion of bisimulation is introduced for some fragments of XPath$_=$, and a van Benthem like characterization theorem is shown for some of them. In this work we show a definability theorem, which answers the basic question of when a class of data trees is definable by a set of formulas, or by a single formula, over two fragments of XPath$_=$:

the *downward* fragment (which only has the 'child' accessibility relation) and the *vertical* fragment (which has both 'child' and 'parent' axes).

Our main result is the analog of the classic first-order definability theorem (see, e.g.[4, Cor. 6.1.16]), which can be stated as follows:

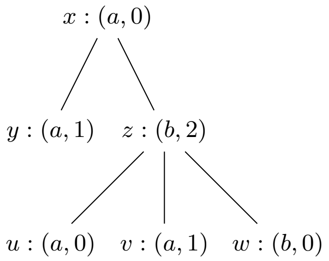
> *A class of models K is definable by means of a set of first-order formulas if and only if K is closed under ultraproducts and isomorphisms, and the complement of K is closed under ultrapowers. Also K is definable by a single first-order formula if and only if both K and its complement are closed under ultraproducts and isomorphisms.*

The above result was adapted to the context of many modal logics, where the notion of *isomorphism* is replaced by the weaker concept of *bisimulation* (the one which turns to be adequate for the chosen modal logic). Thus definability theorems were established for the basic modal logic [6], for temporal logics with *since* and *until* operators [11], for negation-free modal languages [12], etc. A global counterpart was studied in [7], and a general framework stating sufficient conditions for an arbitrary (modal) logic $\mathcal{L}$ to verify it was given in [1]. One of those requirements is that the models of $\mathcal{L}$ are closed under ultraproducts, which is true for the aforementioned logics, but not for XPath$_=$: models of XPath$_=$ are data trees, which may not remain connected under ultraproducts. Hence one cannot expect to use that framework in this case.

Though we take as motivation the current relevance of XML documents (which of course are finite) and the logics for reasoning over them, we do not restrict ourselves to the finite case. Indeed, an infinite set of formulas may force all its data tree models to be infinite. Hence, since we aim at working with arbitrary sets of formulas, we must consider arbitrary (i.e. finite or infinite) data trees.

Our definability theorems for XPath$_=$ themselves are shown using rather known techniques. The main contribution, however, is to devise and calibrate the adequate notions to be used in the XPath$_=$ scenario, and to study the subtle interaction between them:

- *Bisimulation*: already introduced in [8], it is the counterpart of *isomorphisms* in the classical theorem for first-order logic. In [8] it is shown that if two (possibly infinite) data trees are bisimilar then they are logically equivalent (that is, they are not distinguishable by an XPath$_=$ formula) but that the converse is not true in general.
- *Saturation*: we define and study the new notion of *XPath$_=$-saturation*. We show that for XPath$_=$-saturated data trees being bisimilar is the same as being logically equivalent. It is also shown that a 2-saturated data tree (regarded as a first-order structure) is already XPath$_=$-saturated.
- *Utlraproducts*: contrary to other adaptations of the classical first-order definability theorem to modal logics, in our case we have to adjust also the notion of *ultraproduct*, and so we work with a variant of it called *quasi-ultraproduct*. The reason is that we must not abandon the universe of data trees, as these are the only allowed models of XPath$_=$.



- $\varphi_1 = \langle\downarrow[a]\rangle \wedge \langle\downarrow[b]\rangle$, $[\![\varphi_1]\!]^{\mathcal{T}} = \{x, z\}$ ("*nodes with a child labeled a and a child labeled b*")
- $\varphi_2 = \langle\downarrow[a] \neq \downarrow[a]\rangle$, $[\![\varphi_2]\!]^{\mathcal{T}} = \{z\}$ ("*nodes with two children with same tag a but different data value*")
- $\varphi_3 = \langle\varepsilon \neq \uparrow[\langle\varepsilon \neq \uparrow\rangle]\rangle$, $[\![\varphi_3]\!] = \{u, v, w\}$ ("*nodes with a data value different from the one of his parent, who, in turn, has a data value different from his parent*")
- $\varphi_4 = \langle\varepsilon \neq \downarrow\downarrow[\varphi_3]\rangle$, $[\![\varphi_4]\!] = \{x\}$ ("*nodes with a downward path of length 2, with all distinct data values*")

Fig. 1. A data tree $\mathcal{T} \in \mathit{Trees}(\{a, b\} \times \mathbb{N})$ and the meaning of some XPath$^{\updownarrow}_{=}$-formulas

There are many works in the literature studying the expressive power of Core-XPath (see e.g. [10,13,14]). All these consider the navigational fragment of XPath. A first step towards the study of the expressive power of XPath when equipped with (in)equality test over data trees, is the recent paper [8]. We aim to shed more light in this direction.

The paper is organized as follows, In §2 we introduce the formal syntax and semantics of the downward and vertical fragments of XPath$_=$, together with notions of bisimulations from [8]. Suitable notions of saturation for both fragments are given in §3, where it is also shown that for saturated trees bisimilarity coincides with logical equivalence. In §4 we explain the connection between XPath$_=$ and first-order logic, and we introduce the idea of quasi-ultraproducts for the downward and vertical fragments. In §5 we state the theorems on definability, and we close in §6 with a few words about future research and show some applications of the definability results.

Acknowledgements. This work was partially supported by grant ANPCyT-PICT-2011-0365, UBACyT 20020110100025, the FP7-PEOPLE-2011-IRSES Project MEALS and the Laboratoire International Associé INFINIS.

2 Preliminaries

Data trees. Let *Trees*(A) be the set of ordered and unranked (finite or infinite) trees over an arbitrary alphabet A. We say that $\mathcal{T}$ is a **data tree** if it is a tree from *Trees*$(\mathbb{A} \times \mathbb{D})$, where $\mathbb{A}$ is a finite set of **labels** and $\mathbb{D}$ is an infinite set of **data values** (see Figure 1 for an example). A data tree is **finitely branching** if every node has finitely many children. For any given data tree $\mathcal{T}$, we denote by T its set of nodes. We use letters x, y, z, u, v, w as variables for nodes. Given a node $x \in T$ of $\mathcal{T}$, we write $\mathit{label}(x) \in \mathbb{A}$ to denote the node's label, and $\mathit{data}(x) \in \mathbb{D}$ to denote the node's data value.

Given two nodes $x, y \in T$ we write $x \to y$ if y is a child of x, and $x \overset{n}{\to} y$ if y is a descendant of x at distance n. In particular, $\overset{1}{\to}$ is the same as $\to$, and $\overset{0}{\to}$ is the identity relation. $(\overset{n}{\to} y)$ denotes the sole ancestor of y at distance n (assuming it has one).

Vertical and Downward XPath with data tests. We work with a simplification of XPath, stripped of its syntactic sugar. We consider fragments of XPath that correspond to the navigational part of XPath 1.0 with data equality and inequality. $\text{XPath}_=$ is a two-sorted language, with **path expressions** (that we write α, β, γ) and **node expressions** (that we write φ, ψ, η). The **vertical XPath**, notated $\text{XPath}_=^{\updownarrow}$ is defined by mutual recursion as follows:

$$\begin{array}{lll} \alpha, \beta ::= o \mid [\varphi] \mid \alpha\beta \mid \alpha \cup \beta & & o \in \{\varepsilon, \uparrow, \downarrow\} \\ \varphi, \psi ::= a \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \langle\alpha\rangle \mid \langle\alpha = \beta\rangle \mid \langle\alpha \neq \beta\rangle & & a \in \mathbb{A} \end{array}$$

We call **downward XPath**, notated $\text{XPath}_=^{\downarrow}$, to the syntactic fragment which only uses the navigation axis $\downarrow$, but not $\uparrow$. An **$\text{XPath}_=^{\updownarrow}$-formula** [resp. **$\text{XPath}_=^{\downarrow}$-formula**] is simply a node expression of $\text{XPath}_=^{\updownarrow}$ [resp. $\text{XPath}_=^{\downarrow}$].

Semantics of $\text{XPath}_=^{\updownarrow}$ in a given data tree $\mathcal{T}$ are defined as follows:

$$\begin{array}{lll} [\![\downarrow]\!]^{\mathcal{T}} = \{(x,y) \mid x \to y\} & [\![\neg\varphi]\!]^{\mathcal{T}} = T \setminus [\![\varphi]\!]^{\mathcal{T}} & [\![a]\!]^{\mathcal{T}} = \{x \in T \mid \mathit{label}(x) = a\} \\ [\![\uparrow]\!]^{\mathcal{T}} = \{(x,y) \mid y \to x\} & [\![\varphi \wedge \psi]\!]^{\mathcal{T}} = [\![\varphi]\!]^{\mathcal{T}} \cap [\![\psi]\!]^{\mathcal{T}} & [\![[\varphi]]\!]^{\mathcal{T}} = \{(x,x) \mid x \in [\![\varphi]\!]^{\mathcal{T}}\} \\ [\![\varepsilon]\!]^{\mathcal{T}} = \{(x,x) \mid x \in T\} & [\![\alpha \cup \beta]\!]^{\mathcal{T}} = [\![\alpha]\!]^{\mathcal{T}} \cup [\![\beta]\!]^{\mathcal{T}} & \end{array}$$

$$\begin{aligned} [\![\alpha\beta]\!]^{\mathcal{T}} &= \{(x,z) \mid (\exists y \in T)\ (x,y) \in [\![\alpha]\!]^{\mathcal{T}}, (y,z) \in [\![\beta]\!]^{\mathcal{T}}\} \\ [\![\langle\alpha\rangle]\!]^{\mathcal{T}} &= \{x \in T \mid (\exists y \in T)\ (x,y) \in [\![\alpha]\!]^{\mathcal{T}}\} \\ [\![\langle\alpha = \beta\rangle]\!]^{\mathcal{T}} &= \{x \in T \mid (\exists y,z \in T)(x,y) \in [\![\alpha]\!]^{\mathcal{T}}, (x,z) \in [\![\beta]\!]^{\mathcal{T}}, \mathit{data}(y) = \mathit{data}(z)\} \\ [\![\langle\alpha \neq \beta\rangle]\!]^{\mathcal{T}} &= \{x \in T \mid (\exists y,z \in T)(x,y) \in [\![\alpha]\!]^{\mathcal{T}}, (x,z) \in [\![\beta]\!]^{\mathcal{T}}, \mathit{data}(y) \neq \mathit{data}(z)\} \end{aligned}$$

See Figure 1 for the semantics of some formulas. For a data tree $\mathcal{T}$ and $u \in T$, we write $\mathcal{T}, u \models \varphi$ to denote $u \in [\![\varphi]\!]^{\mathcal{T}}$, and we say that $\mathcal{T}, u$ satisfies φ or that φ is true at $\mathcal{T}, u$. Let $\text{Th}_{\updownarrow}(\mathcal{T}, u)$ [resp. $\text{Th}_{\downarrow}(\mathcal{T}, u)$] be the set of all $\text{XPath}_=^{\updownarrow}$-formulas [resp. $\text{XPath}_=^{\downarrow}$-formulas] true at $\mathcal{T}, u$. In terms of expressive power, it is easy to see that $\cup$ is unessential (see [8, §2.2]). We will henceforth assume that formulas do not contain union of path expressions.

Let $\mathcal{T}$ and $\mathcal{T}'$ be data trees, and let $u \in T$, $u' \in T'$. We say that $\mathcal{T}, u$ and $\mathcal{T}', u'$ are **equivalent for $\text{XPath}_=^{\updownarrow}$** [resp. **equivalent for $\text{XPath}_=^{\downarrow}$**] (notation: $\mathcal{T}, u \equiv^{\updownarrow} \mathcal{T}', u'$ [resp. $\mathcal{T}, u \equiv^{\downarrow} \mathcal{T}', u'$]) iff for all formulas $\varphi \in \text{XPath}_=^{\updownarrow}$ [resp. $\varphi \in \text{XPath}_=^{\downarrow}$], we have $\mathcal{T}, u \models \varphi$ iff $\mathcal{T}', u' \models \varphi$.

Bisimulations. In [8] the notions of downward and vertical bisimulations are introduced. We reproduce them here, as they are key concepts for our results.

We say that $u \in T$ and $u' \in T'$ are **bisimilar for $\text{XPath}_=^{\downarrow}$** (or **$\downarrow$-bisimilar**; notation: $\mathcal{T}, u \mathrel{\underline{\leftrightarrow}}^{\downarrow} \mathcal{T}', u'$) iff there is a relation $Z \subseteq T \times T'$ such that uZu' and for all $x \in T$ and $x' \in T'$ we have

- **Harmony:** If xZx' then $\mathit{label}(x) = \mathit{label}(x')$.
- **Zig:** If xZx', $x \overset{n}{\to} v$ and $x \overset{m}{\to} w$ then there are $v', w' \in T'$ such that $x' \overset{n}{\to} v'$, $x' \overset{m}{\to} w'$ and
  1. $\mathit{data}(v) = \mathit{data}(w) \Leftrightarrow \mathit{data}(v') = \mathit{data}(w')$,
  2. $(\overset{i}{\to} v)\, Z\, (\overset{i}{\to} v')$ for all $0 \leq i < n$, and
  3. $(\overset{i}{\to} w)\, Z\, (\overset{i}{\to} w')$ for all $0 \leq i < m$.
- **Zag:** If xZx', $x' \overset{n}{\to} v'$ and $x' \overset{m}{\to} w'$ then there are $v, w \in T$ such that $x \overset{n}{\to} v$, $x \overset{m}{\to} w$ and items 1, 2 and 3 above are verified.

We say that $u \in T$ and $u' \in T'$ are **bisimilar for XPath$^{\updownarrow}_{=}$** (or **$\updownarrow$-bisimilar**; notation: $\mathcal{T}, u \,\underline{\leftrightarrow}^{\updownarrow}\, \mathcal{T}', u'$) iff there is a relation $Z \subseteq T \times T'$ such that uZu' and for all $x \in T$ and $x' \in T'$ we have

- **Harmony:** If xZx' then $\mathit{label}(x) = \mathit{label}(x')$,
- **Zig:** If xZx', $y \overset{n}{\to} x$ and $y \overset{m}{\to} z$ then there are $y', z' \in T'$ such that $y' \overset{n}{\to} x'$, $y' \overset{m}{\to} z'$, $\mathit{data}(z) = \mathit{data}(x) \Leftrightarrow \mathit{data}(z') = \mathit{data}(x')$, and zZz'.
- **Zag:** If xZx', $y' \overset{n}{\to} x'$ and $y' \overset{m}{\to} z'$ then there are $y, z \in T$ such that $y \overset{n}{\to} x$, $y \overset{m}{\to} z$, $\mathit{data}(z) = \mathit{data}(x) \Leftrightarrow \mathit{data}(z') = \mathit{data}(x')$, and zZz'.

The main results establishing the connection between bisimulation and equivalence is the following:

Theorem 1 ([8]). *If $\mathcal{T}, u \,\underline{\leftrightarrow}^{\downarrow}\, \mathcal{T}', u'$ then $\mathcal{T}, u \equiv^{\downarrow} \mathcal{T}', u'$, and if $\mathcal{T}, u \,\underline{\leftrightarrow}^{\updownarrow}\, \mathcal{T}', u'$, then $\mathcal{T}, u \equiv^{\updownarrow} \mathcal{T}', u'$.*

3 Saturation

In [8] is is shown that the reverse implications of Theorem 1 hold over finitely branching trees. However, they do not hold in general. In this section we introduce notions of saturation for the downward and vertical fragments of XPath, and show that the reverse implications of Theorem 1 are true over saturated data trees.

Saturation for the downward fragment. Let $\langle \Sigma_1, \ldots, \Sigma_n \rangle$ and $\langle \Gamma_1, \ldots, \Gamma_m \rangle$ be tuples of sets of XPath$^{\downarrow}_{=}$-formulas. Given a data tree $\mathcal{T}$ and $u \in T$, we say that $\langle \Sigma_1, \ldots, \Sigma_n \rangle$ and $\langle \Gamma_1, \ldots, \Gamma_m \rangle$ are **$=^{\downarrow}_{n,m}$-satisfiable** [resp. **$\neq^{\downarrow}_{n,m}$-satisfiable**] at $\mathcal{T}, u$ if there exist $v_0 \to v_1 \to \cdots \to v_n \in T$ and $w_0 \to w_1 \to \cdots \to w_m \in T$ such that $u = v_0 = w_0$ and

1. for all $i \in \{1, \ldots, n\}$, $\mathcal{T}, v_i \models \Sigma_i$;
2. for all $j \in \{1, \ldots, m\}$, $\mathcal{T}, w_j \models \Gamma_j$; and
3. $\mathit{data}(v_n) = \mathit{data}(w_m)$ [resp. $\mathit{data}(v_n) \neq \mathit{data}(w_m)$].

We say that $\langle \Sigma_1, \ldots, \Sigma_n \rangle$ and $\langle \Gamma_1, \ldots, \Gamma_m \rangle$ are **$=^{\downarrow}_{n,m}$-finitely satisfiable** [resp. **$\neq^{\downarrow}_{n,m}$-finitely satisfiable**] at $\mathcal{T}, u$ if for every finite $\Sigma'_i \subseteq \Sigma_i$ and finite $\Gamma'_j \subseteq \Gamma_j$, we have that $\langle \Sigma'_1, \ldots, \Sigma'_n \rangle$ and $\langle \Gamma'_1, \ldots, \Gamma'_m \rangle$ are $=^{\downarrow}_{n,m}$-satisfiable [resp. $\neq^{\downarrow}_{n,m}$-satisfiable] at $\mathcal{T}, u$.

Definition 2. *We say that a data tree $\mathcal{T}$ is* $\downarrow$***-saturated*** *if for every $n, m \in \mathbb{N}$, every pair of tuples $\langle \varSigma_1, \dots, \varSigma_n \rangle$ and $\langle \varGamma_1, \dots, \varGamma_m \rangle$ of sets of XPath$^{\downarrow}_{=}$-formulas, every $u \in T$, and $\star \in \{=, \neq\}$, the following is true:*

> *if $\langle \varSigma_1, \dots, \varSigma_n \rangle$ and $\langle \varGamma_1, \dots, \varGamma_m \rangle$ are $\star^{\downarrow}_{n,m}$-finitely satisfiable at $\mathcal{T}, u$ then $\langle \varSigma_1, \dots, \varSigma_n \rangle$ and $\langle \varGamma_1, \dots, \varGamma_m \rangle$ are $\star^{\downarrow}_{n,m}$-satisfiable at $\mathcal{T}, u$.*

Proposition 3. *Any finitely branching data tree is $\downarrow$-saturated.*

Proof. Suppose by contradiction that there is $u \in T$ and tuples $\langle \varSigma_1, \dots, \varSigma_n \rangle$ and $\langle \varGamma_1, \dots, \varGamma_m \rangle$ of sets of XPath$^{\downarrow}_{=}$-formulas which are finitely $=^{\downarrow}_{n,m}$-satisfiable at $\mathcal{T}, u$ but not $=^{\downarrow}_{n,m}$-satisfiable at $\mathcal{T}, u$ (the case for $\mathcal{T}$ being $\neq^{\downarrow}_{n,m}$-satisfiable is analogous). Let

$$P = \{(v, w) \in T^2 \mid u \overset{n}{\to} v \wedge u \overset{m}{\to} w \wedge data(v) = data(w)\}.$$

Observe that P is finite because $\mathcal{T}$ is finitely branching. It is clear that if $(v, w) \in P$, so that $u = v_0 \to v_1 \to \dots \to v_n = v \in T$, and $u = w_0 \to w_1 \to \dots \to w_m = w \in T$ then either

1. there is $i \in \{1, \dots, n\}$ such that $\mathcal{T}, v_i \not\models \varSigma_i$, or
2. there is $j \in \{1, \dots, m\}$ such that $\mathcal{T}, w_j \not\models \varGamma_j$.

We will define sets $(\varSigma_{i,v,w})_{1 \le i \le n}$ and $(\varGamma_{j,v,w})_{1 \le j \le m}$, each one of them with at most one element, as follows: If case 1 holds, assume i_0 is the least such number and define $\varSigma_{i_0,v,w}$ as $\{\rho\}$ for some formula $\rho \in \varSigma_{i_0}$ such that $\mathcal{T}, v_{i_0} \not\models \rho$, define $\varSigma_{i,v,w} = \emptyset$ for any $i \in \{1, \dots, n\} \setminus \{i_0\}$, and define $\varGamma_{j,v,w} = \emptyset$ for any $j \in \{1, \dots, m\}$. If case 1 does not hold then case 2 holds, so assume j_0 is the least such number and define $\varGamma_{j_0,v,w}$ as $\{\rho\}$ for some formula $\rho \in \varGamma_{j_0}$ such that $\mathcal{T}, w_{j_0} \not\models \rho$, define $\varGamma_{j,v,w} = \emptyset$ for any $j \in \{1, \dots, m\} \setminus \{j_0\}$, and define $\varSigma_{i,v,w} = \emptyset$ for any $i \in \{1, \dots, n\}$. Finally, define the finite sets $\varSigma'_i = \bigcup_{(v,w) \in P} \varSigma_{i,v,w}$ and $\varGamma'_j = \bigcup_{(v,w) \in P} \varGamma_{j,v,w}$. By construction, we have $\varSigma'_i \subseteq \varSigma_i$, $\varGamma'_j \subseteq \varGamma_j$ and $\langle \varSigma'_1, \dots, \varSigma'_n \rangle$ and $\langle \varGamma'_1, \dots, \varGamma'_m \rangle$ are not $=^{\downarrow}_{n,m}$-satisfiable at $\mathcal{T}, u$ which is a contradiction. □

Proposition 4. *Let $\mathcal{T}$ and $\mathcal{T}'$ be $\downarrow$-saturated data trees, and let $u \in T$ and $u' \in T'$. If $\mathcal{T}, u \equiv^{\downarrow} \mathcal{T}', u'$, then $\mathcal{T}, u \leftrightarroweq^{\downarrow} \mathcal{T}', u'$.*

Proof. We show that Z, defined by xZx' iff $\mathcal{T}, x \equiv^{\downarrow} \mathcal{T}', x'$ is a $\downarrow$-bisimulation between $\mathcal{T}, u$ and $\mathcal{T}', u'$. Clearly uZu', and Harmony holds. We only need to show that Zig and Zag are satisfied. We see only Zig, as Zag is analogous.

Suppose xZx', $x = v_0 \to v_1 \to \dots \to v_n$ and $x = w_0 \to w_1 \to \dots \to w_m$ are paths on $\mathcal{T}$, and $data(v_n) = data(w_m)$ (the case $data(v_n) \neq data(w_m)$ is shown analogously). For $i \in \{1, \dots, n\}$, let $\varSigma_i = \mathrm{Th}_{\downarrow}(\mathcal{T}, v_i)$, and for $j \in \{1, \dots, m\}$, let $\varGamma_j = \mathrm{Th}_{\downarrow}(\mathcal{T}, w_j)$. Furthermore, let $\varSigma'_i$ be a finite subset of $\varSigma_i$, and let $\varGamma'_j$ be a finite subset of $\varGamma_j$. Define

$$\varphi = \langle \downarrow [\wedge \varSigma'_1] \downarrow \dots \downarrow [\wedge \varSigma'_n] = \downarrow [\wedge \varGamma'_1] \downarrow \dots \downarrow [\wedge \varGamma'_m] \rangle.$$

It is clear that $\mathcal{T}, x \models \varphi$, and since by definition of Z we have $\mathcal{T}, x \equiv^{\downarrow} \mathcal{T}', x'$, we conclude that $\mathcal{T}', x' \models \varphi$. Hence $\langle \Sigma_1', \dots, \Sigma_n' \rangle$ and $\langle \Gamma_1', \dots, \Gamma_m' \rangle$ are $=^{\downarrow}_{n,m}$-satisfiable at x'. This holds for *any* finite sets $\Sigma_i' \subseteq \Sigma_i$ and $\Gamma_j' \subseteq \Gamma_j$, and so $\langle \Sigma_1, \dots, \Sigma_n \rangle$ and $\langle \Gamma_1, \dots, \Gamma_m \rangle$ are $=^{\downarrow}_{n,m}$-finitely satisfiable at x', Since $\mathcal{T}'$ is $\downarrow$-saturated, $\langle \Sigma_1, \dots, \Sigma_n \rangle$ and $\langle \Gamma_1, \dots, \Gamma_m \rangle$ are $=^{\downarrow}_{n,m}$-satisfiable at $\mathcal{T}', x'$, so there are paths $x' = v_0' \to v_1' \to \cdots \to v_n'$ and $x' = w_0' \to w_1' \to \cdots \to w_m'$ on $\mathcal{T}'$ such that

i. $data(v_n') = data(w_m')$;
ii. for all $1 \leq i \leq n$, $\mathcal{T}', v_i' \models \mathrm{Th}_{\downarrow}(\mathcal{T}, v_i)$, i.e. $\mathcal{T}, v_i \equiv^{\downarrow} \mathcal{T}', v_i'$; and
iii. for all $1 \leq j \leq m$, $\mathcal{T}', w_j' \models \mathrm{Th}_{\downarrow}(\mathcal{T}, w_j)$, i.e $\mathcal{T}, w_j \equiv^{\downarrow} \mathcal{T}', w_j'$.

By the definition of Z, conditions i, ii and iii above imply items 1, 2 and 3 of the Zig clause for $\downarrow$-bisimulation. □

Saturation for the vertical fragment. Given a data tree $\mathcal{T}$ and $u \in T$, we say that the set of XPath$^{\updownarrow}_{=}$-formulas Γ is $=^{\updownarrow}_{n,m}$**-satisfiable** [resp. $\neq^{\updownarrow}_{n,m}$**-satisfiable**] at $\mathcal{T}, u$ if there exist $v, w \in T$ such that $v \xrightarrow{n} u$, $v \xrightarrow{m} w$, $w \models \Gamma$ and $data(u) = data(w)$ [resp. $data(u) \neq data(w)$]. We say that Γ is $=^{\updownarrow}_{n,m}$**-finitely satisfiable** [resp. $\neq^{\updownarrow}_{n,m}$**-finitely satisfiable**] at $\mathcal{T}, u$ if for every finite $\Gamma' \subseteq \Gamma$, we have that Γ' is $=^{\updownarrow}_{n,m}$-satisfiable [resp. $=^{\updownarrow}_{n,m}$-satisfiable] at $\mathcal{T}, u$.

Definition 5. *We say that a data tree $\mathcal{T}$ is* $\updownarrow$***-saturated*** *if for every set of XPath$^{\updownarrow}_{=}$-formulas Γ, every $u \in T$, every $n, m \in \mathbb{N}$, and $\star \in \{=, \neq\}$, the following is true:*

if Γ is $\star^{\updownarrow}_{n,m}$-finitely satisfiable at $\mathcal{T}, u$ then Γ is $\star^{\updownarrow}_{n,m}$-satisfiable at $\mathcal{T}, u$.

Proposition 6. *Let $\mathcal{T}$ and $\mathcal{T}'$ be $\updownarrow$-saturated data trees, and let $u \in T$ and $u' \in T'$. If $\mathcal{T}, u \equiv^{\updownarrow} \mathcal{T}', u'$, then $\mathcal{T}, u \,\underline{\leftrightarrow}^{\updownarrow}\, \mathcal{T}', u'$.*

Proof. We show that $Z \subseteq T \times T'$, defined by xZx' iff $\mathcal{T}, x \equiv^{\updownarrow} \mathcal{T}', x'$ is a $\updownarrow$-bisimulation between $\mathcal{T}, u$ and $\mathcal{T}', u'$. Clearly uZu', and Harmony also holds, so we only need to show that Zig and Zag are satisfied. We see only Zig, as Zag is analogous.

Suppose xZx', $y \xrightarrow{n} x$ and $y \xrightarrow{m} z$ are in $\mathcal{T}$, and $data(x) = data(z)$ (the case $data(x) \neq data(z)$ can be shown analogously). Let $\Gamma = \mathrm{Th}_{\updownarrow}(\mathcal{T}, z)$, and let Γ' be a finite subset of Γ. Define

$$\varphi = \langle \varepsilon = \uparrow^{n} \downarrow^{m} [\wedge \Gamma'] \rangle.$$

It is clear that $\mathcal{T}, x \models \varphi$, and since by definition of Z we have $\mathcal{T}, x \equiv^{\updownarrow} \mathcal{T}', x'$, we conclude that $\mathcal{T}', x' \models \varphi$. Hence Γ' is $=^{\updownarrow}_{n,m}$-satisfiable at x'. This holds for *any* finite set $\Gamma' \subseteq \Gamma$, and so Γ is $=^{\updownarrow}_{n,m}$-finitely satisfiable at x'. Since $\mathcal{T}'$ is $\updownarrow$-saturated, Γ is $=^{\updownarrow}_{n,m}$-satisfiable at x', and thus there are $y' \xrightarrow{n} x'$ and $y' \xrightarrow{m} z'$ on $\mathcal{T}'$ such that $data(x') = data(z')$ and $\mathcal{T}', z' \models \mathrm{Th}_{\updownarrow}(\mathcal{T}, z)$, i.e $\mathcal{T}, z \equiv^{\updownarrow} \mathcal{T}', z'$. By the definition of Z, we have zZz' and hence the Zig clause for $\updownarrow$-bisimulation is verified. □

4 Weak Data Trees and Quasi-ultraproducts

We fix the signature σ with binary relations $\rightsquigarrow$ and $\sim$, and a unary predicate P_a for each $a \in \mathbb{A}$. Any data tree $\mathcal{T}$ can be seen as a first-order σ-structure, where

$$\begin{aligned} \rightsquigarrow^{\mathcal{T}} &= \{(x,y) \in T^2 \mid x \to y \text{ in } \mathcal{T}\}; \\ \sim^{\mathcal{T}} &= \{(x,y) \in T^2 \mid \mathit{data}(x) = \mathit{data}(y)\}; \\ P_a^{\mathcal{T}} &= \{x \in T \mid \mathit{label}(x) = a\}. \end{aligned}$$

If $\varphi(x)$ is a first-order formula with a free variable x, we use $\mathcal{T} \models \varphi[a]$, as usual, to denote that that φ is true in $\mathcal{T}$ under the valuation which maps x to $a \in T$. In [8] it is shown a truth preserving translation Tr_x mapping XPath$^{\updownarrow}_{=}$-formulas into first-order σ-formulas with one free variable x. By *truth preserving* we mean that for $\varphi \in$ XPath$^{\updownarrow}_{=}$ we have $\mathcal{T}, u \models \varphi$ iff $\mathcal{T} \models \mathrm{Tr}_x(\varphi)[u]$.

For reasons that will become clearer later on, we will need to work with σ-structures which are slightly more general than data trees.

Definition 7. *A σ-structure $\mathcal{T}$ is a* ***weak data tree*** *if $\sim$ is an equivalence relation; there is exactly one node r with no u such that $u \rightsquigarrow r$ (r is called* root *of $\mathcal{T}$); for all nodes $x \neq r$ there is exactly one y such that $y \rightsquigarrow x$; and for each $n \geq 0$ the relation $\rightsquigarrow$ has no cycles of length n.*

Observe that a weak data tree need not be connected, and that the class of weak data trees is elementary, i.e. definable by a set of first-order σ-sentences (with equality). For a weak data tree $\mathcal{T}$ and $u \in T$, let $\mathcal{T}|u$ denote the substructure of $\mathcal{T}$ induced by $\{v \in T \mid u \rightsquigarrow^* v\}$. Observe that in this case $\mathcal{T}|u$ is a data tree.

The following proposition shows the 'local' aspect of XPath$^{\downarrow}_{=}$ and XPath$^{\updownarrow}_{=}$. It is stated in terms of first-order because models are weak data trees.

Proposition 8. *Let $\mathcal{T}$ be a weak data tree and let $r \rightsquigarrow^* u$ in $\mathcal{T}$.*

1. *If $\varphi \in$ XPath$^{\downarrow}_{=}$-formula then $\mathcal{T} \models \mathrm{Tr}_x(\varphi)[u]$ iff $\mathcal{T}|r \models \mathrm{Tr}_x(\varphi)[u]$.*
2. *If r is the root of $\mathcal{T}$ and $\varphi \in$ XPath$^{\updownarrow}_{=}$ then $\mathcal{T} \models \mathrm{Tr}_x(\varphi)[u]$ iff $\mathcal{T}|r \models \mathrm{Tr}_x(\varphi)[u]$.*

Observe that the condition of r being the root in the second item is needed. Suppose for example we are on the data tree with only 2 nodes, the root r and its child u, with same data value. Consider now $\varphi = \langle \varepsilon = \uparrow \rangle$. Clearly $\mathcal{T} \models \mathrm{Tr}_x(\varphi)[u]$, but $\mathcal{T}|u \not\models \mathrm{Tr}_x(\varphi)[u]$.

If $\mathcal{M}$ is a first-order σ-structure and $A \subseteq M$, we denote by σ_A the language obtained by adding to σ constant symbols for each $a \in A$. $\mathcal{M}$ can be seen as a σ_A structure by interpreting the new symbols in the obvious way. Let $\mathrm{Th}_A(\mathcal{M})$ be the set of all σ_A-sentences true in $\mathcal{M}$. Let κ be a cardinal. We recall the definition of κ-saturated first-order structures. We say that the σ-structure $\mathcal{M}$ is κ-saturated if for all $A \subseteq M$ and all n, if $|A| < \kappa$ and $\Gamma(x_1, \dots, x_n)$ is a set of σ_A-formulas with free variables among $x_1, \dots, x_n$ such that $\Gamma(x_1, \dots, x_n) \cup \mathrm{Th}_A(\mathcal{M})$ is satisfiable, then $\Gamma(x_1, \dots, x_n)$ is realized in $\mathcal{M}$.

We now show that 2-saturated data trees are already both downward and vertical saturated. For technical reasons we state these results in the more general setting of weak data trees.

Proposition 9. *Let $\mathcal{T}$ be a 2-saturated weak data tree and $r \in T$.*

1. *$\mathcal{T}|r$ is a $\downarrow$-saturated data tree.*
2. *If r is the root of T then $\mathcal{T}|r$ is a $\updownarrow$-saturated data tree.*

Proof. Let $\mathcal{T}' = \mathcal{T}|r$ and let $u \in T'$. For item 1, let $\langle \Sigma_1, \ldots, \Sigma_n\rangle$ and $\langle \Gamma_1, \ldots, \Gamma_m\rangle$ be tuples of sets of $\mathrm{XPath}^{\downarrow}_{=}$-formulas. Suppose $\langle \Sigma_1, \ldots, \Sigma_n\rangle$ and $\langle \Gamma_1, \ldots, \Gamma_m\rangle$ are $=^{\downarrow}_{n,m}$-finitely satisfiable at $\mathcal{T}', u$ (the case for $\neq^{\downarrow}_{n,m}$-finitely satisfiable is analogous). We show that $\langle \Sigma_1, \ldots, \Sigma_n\rangle$ and $\langle \Gamma_1, \ldots, \Gamma_m\rangle$ are $=^{\downarrow}_{n,m}$-satisfiable at $\mathcal{T}', u$. Consider the following first-order $\sigma_{\{u\}}$-formula with free variables $\bar{x} = x_1, \ldots, x_n$ and $\bar{y} = y_1, \ldots, y_m$:

$$\varphi(\bar{x}, \bar{y}) = u \rightsquigarrow x_1 \wedge \bigwedge_{i=1}^{n-1} x_i \rightsquigarrow x_{i+1} \wedge u \rightsquigarrow y_1 \wedge \bigwedge_{j=1}^{m-1} y_j \rightsquigarrow y_{j+1} \wedge x_n \sim y_m .$$

Define the following set of first-order $\sigma_{\{u\}}$-formulas:

$$\Delta(\bar{x}, \bar{y}) = \{\varphi(\bar{x}, \bar{y})\} \cup \bigcup_{i=1}^{n} \mathrm{Tr}_{x_i}(\Sigma_i) \cup \bigcup_{j=1}^{m} \mathrm{Tr}_{y_j}(\Gamma_j).$$

Let $\Delta'(\bar{x}, \bar{y})$ be a finite subset of $\Delta(\bar{x}, \bar{y})$. Since $\langle \Sigma_1, \ldots, \Sigma_n\rangle$ and $\langle \Gamma_1, \ldots, \Gamma_m\rangle$ are $=^{\downarrow}_{n,m}$-finitely satisfiable at $\mathcal{T}', u$, then $\Delta'(\bar{x}, \bar{y})$ is satisfiable and, by item 1 of Proposition 8, consistent with $\mathrm{Th}_{\{u\}}(\mathcal{T})$. By compactness, $\Delta(\bar{x}, \bar{y})$ is satisfiable and consistent with $\mathrm{Th}_{\{u\}}(\mathcal{T})$. By 2-saturation, we conclude that $\Delta(\bar{x}, \bar{y})$ is realizable in $\mathcal{T}$, say at $\bar{v} = v_1, \ldots, v_n$ and $\bar{w} = w_1, \ldots, w_m$. Thus we have:

i. $u \rightsquigarrow v_1 \rightsquigarrow \cdots \rightsquigarrow v_n$ and $u \rightsquigarrow w_1 \rightsquigarrow \cdots \rightsquigarrow w_m$ in $\mathcal{T}$, and hence in $\mathcal{T}'$;
ii. for all $i \in \{1, \ldots, n\}$, $\mathcal{T} \models \mathrm{Tr}_{x_i}(\Sigma_i)[v_i]$, and for all $j \in \{1, \ldots, m\}$, $\mathcal{T} \models \mathrm{Tr}_{y_j}(\Gamma_j)[w_j]$; by item 1 of Proposition 8 this implies that $\mathcal{T}' \models \mathrm{Tr}_{x_i}(\Sigma_i)[v_i]$ and $\mathcal{T}' \models \mathrm{Tr}_{y_j}(\Gamma_j)[w_j]$;
iii. $v_n \sim w_m$ in $\mathcal{T}$, and hence in $\mathcal{T}'$.

Since Tr is truth preserving, we have that for all $i \in \{1, \ldots, n\}$, $\mathcal{T}', v_i \models \Sigma_i$, and for all $j \in \{1, \ldots, m\}$, $\mathcal{T}', w_i \models \Gamma_i$. Together with i and iii we conclude that $\langle \Sigma_1, \ldots, \Sigma_n\rangle$ and $\langle \Gamma_1, \ldots, \Gamma_m\rangle$ are $=^{\downarrow}_{n,m}$-satisfiable at $\mathcal{T}', u$.

For item 2, let Γ be a set of $\mathrm{XPath}^{\updownarrow}_{=}$-formulas. Suppose Γ is $=^{\updownarrow}_{n,m}$-finitely satisfiable at $\mathcal{T}', u$ (the case for $\neq^{\updownarrow}_{n,m}$-finitely satisfiable is analogous). We show that Γ are $=^{\updownarrow}_{n,m}$-satisfiable at $\mathcal{T}', u$.

Consider the following first-order $\sigma_{\{u\}}$-formula with free variable y:

$$\varphi(y) = (\exists x_0 \ldots \exists x_n)(\exists y_0 \ldots \exists y_m)[x_n = u \wedge y = y_m \wedge x_0 = y_0 \wedge$$
$$\bigwedge_{i=0}^{n-1} x_i \rightsquigarrow x_{i+1} \wedge \bigwedge_{j=0}^{m-1} y_j \rightsquigarrow y_{j+1} \wedge x_n \sim y_m]$$

Define the following set of first-order $\sigma_{\{u\}}$-formulas: $\Delta(y) = \{\varphi(y)\} \cup \mathrm{Tr}_y(\Gamma)$. Let $\Delta'(y)$ be a finite subset of $\Delta(y)$. Since Γ is $=^{\updownarrow}_{n,m}$-finitely satisfiable at $\mathcal{T}', u$, then

$\Delta'(y)$ is satisfiable and, by item 2 of Proposition 8, consistent with $\mathrm{Th}_{\{u\}}(\mathcal{T})$. By compactness, $\Delta(y)$ is satisfiable and consistent with $\mathrm{Th}_{\{u\}}(\mathcal{T})$. By 2-saturation, we conclude that $\Delta(y)$ is realizable in $\mathcal{T}$, say at w. Thus we have:

iv. There is $v \in T$ such that $v \xrightarrow{n} u$ and $v \xrightarrow{m} w$ in $\mathcal{T}$ and hence in $\mathcal{T}'$.
v. $\mathcal{T} \models \mathrm{Tr}_y(\Gamma)[w]$; by item 2 of Proposition 8 this implies that $\mathcal{T}' \models \mathrm{Tr}_y(\Gamma)[w]$;
vi. $u \sim w$ in $\mathcal{T}$, and hence in $\mathcal{T}'$.

Since Tr is truth preserving, we have that $\mathcal{T}', w \models \Gamma$. Together with iv and vi we conclude that Γ is $=^{\updownarrow}_{n,m}$-satisfiable at $\mathcal{T}', u$. □

In what follows, we introduce the notion of quasi-ultraproduct, a variant of the usual notion of first-order model theory, which will be needed for the definability theorems.

Let $I \neq \emptyset$, let U be an ultrafilter over I and let $(\mathcal{T}_i)_{i\in I}$ be a family of data trees. As usual, we denote with $\prod_U \mathcal{T}_i$ the ultraproduct of $(\mathcal{T}_i)_{i\in I}$ modulo U. Observe that by the fundamental theorem of ultraproducts (see e.g. [4, Thm. 4.1.9]), $\prod_U \mathcal{T}_i$ is a weak data tree σ-structure —though it may not be a data tree because it may be disconnected, as it is shown next:

Example 10. For $i \in \mathbb{N}$, let $\mathcal{T}_i$ as any data tree of height at least n, and let u_i as any node of T_i at distance n from the root of $\mathcal{T}_i$. Let $\varphi_n(x)$ be the first-order property "x is at distance at least n from the root". It is clear that $\mathcal{T}_m \models \varphi_n[u_m]$ for every $m \geq n$. Let u^* be the ultralimit of $(u_i)_{i\in I}$ modulo U. Since $\{m \mid m \geq n\} \in U$ for any non-principal U, we conclude that $\prod_U \mathcal{T}_i \models \varphi_n[u^*]$ for every n, and so u^* is disconnected from the root of $\prod_U \mathcal{T}_i$.

Let $(\mathcal{T}_i, u_i)_{i\in I}$ be a family of pointed data trees. The ultraproduct of such *pointed* data trees is defined, as usual, by $(\prod_U \mathcal{T}_i, u^*)$, where u^* is the ultralimit of $(u_i)_{i\in I}$ modulo U.

Definition 11. *Suppose $(\mathcal{T}_i, u_i)_{i\in I}$ is a family of pointed data trees, r_i is the root of $\mathcal{T}_i$, U is an ultrafilter over I, $\mathcal{T}^* = \prod_U \mathcal{T}_i$, and u^* and r^* are the ultralimits of $(u_i)_{i\in I}$ and $(r_i)_{i\in I}$ modulo U respectively.*

1. *The* **$\downarrow$*-quasi ultraproduct*** *of $(\mathcal{T}_i, u_i)_{i\in I}$ modulo U is the pointed data tree $(\mathcal{T}^*|u^*, u^*)$.*
2. *The* **$\updownarrow$*-quasi ultraproduct*** *of $(\mathcal{T}_i, u_i)_{i\in I}$ modulo U is the pair $(\mathcal{T}^*|r^*, u^*)$.*

Observe that both $\mathcal{T}^*|u^*$ and $\mathcal{T}^*|r^*$ are data trees. However, while u^* is in the domain of the former, it may not be in the domain of the latter (cf. Example 10). Hence, in general, pointed data trees are not closed under $\updownarrow$-quasi ultraproduct. Let $k \geq 0$, let $\mathcal{T}$ be a data tree and let $u \in \mathcal{T}$. We say that $(\mathcal{T}, u)$ is a **k-bounded pointed data tree** if u is at distance at most k from the root of $\mathcal{T}$. In particular, if r is the root of $\mathcal{T}$ (as it is often the case) then $(\mathcal{T}, r)$ is a 0-bounded pointed data tree. The following proposition states that k-bounded data trees are closed under $\updownarrow$-quasi ultraproducts.

Proposition 12. *Let $(\mathcal{T}_i, u_i)_{i\in I}$ be a family of k-bounded pointed data trees. Then the $\updownarrow$-quasi ultraproduct of $(\mathcal{T}_i, u_i)_{i\in I}$ is a k-bounded pointed data tree.*

Proof. Let $(\mathcal{T}^{\updownarrow}, u^*)$ be the $\updownarrow$-quasi ultraproduct of $(\mathcal{T}_i, u_i)_{i\in I}$ modulo U. By definition it is clear that $\mathcal{T}^{\updownarrow}$ is a data tree. To see that $u^* \in T^{\updownarrow}$, let

$$\begin{aligned}\varphi(x) = (\exists r)\ &[\neg(\exists y) y \rightsquigarrow r \wedge [r = x \vee r \rightsquigarrow x \vee \\ &\bigvee_{1\leq i<k} (\exists z_1 \ldots \exists z_i)[r \rightsquigarrow z_1 \wedge z_{i-1} \rightsquigarrow x \wedge \bigwedge_{1\leq j<i-1} z_j \rightsquigarrow z_{j+1}]]],\end{aligned}$$

which is a first-order formula for "r is the root and x is at distance at most k from r". Since for every $i \in I$ we have $\mathcal{T}_i \models \varphi[u_i]$, we conclude that $\mathcal{T}^{\updownarrow} \models \varphi[u^*]$ and hence u^* is at distance at most k from the root of $\mathcal{T}^{\updownarrow}$. □

As a particular case one has the notion of **$\downarrow$-quasi ultrapower** and **$\updownarrow$-quasi ultrapower** of a family of pointed data trees. Observe that if $(\mathcal{T}^{\updownarrow}, u^*)$ is the $\updownarrow$-quasi ultrapower of $(\mathcal{T}, u)_{i\in I}$ then u^* belongs to the domain of $\mathcal{T}^{\updownarrow}$ and so $(\mathcal{T}^{\updownarrow}, u^*)$ is a pointed data tree.

5 Definability

In this section we state the main results. If K is a class of pointed data trees, we denote its complement by $\overline{K}$. We begin with the downward fragment.

Lemma 13. *Let $(\mathcal{T}, u)$ and $(\mathcal{T}', u')$ be two pointed data trees such that $\mathcal{T}, u \equiv^{\downarrow} \mathcal{T}', u'$. Then there exist $\downarrow$-quasi ultrapowers $(\mathcal{T}^{\downarrow}, u^*)$ and $(\mathcal{T}'^{\downarrow}, u'^*)$ of $(\mathcal{T}, u)$ and $(\mathcal{T}', u')$ respectively such that $(\mathcal{T}^{\downarrow}, u^*) \,\underline{\leftrightarrow}^{\downarrow}\, (\mathcal{T}'^{\downarrow}, u'^*)$*

Proof. It is known that there is a suitable ultrafilter U such that $\prod_U \mathcal{T}$ and $\prod_U \mathcal{T}'$ are ω-saturated (see e.g. [2, Lem. 2.7.3]). By item 1 Proposition 9, $\mathcal{T}^{\downarrow} = (\prod_U \mathcal{T})|u^*$ and $\mathcal{T}^{\downarrow} = (\prod_U \mathcal{T}')|u'^*$ are $\downarrow$-saturated data trees. By hypothesis $\mathcal{T}, u \equiv^{\downarrow} \mathcal{T}', u'$, and hence $\mathcal{T}^{\downarrow}, u^* \equiv^{\downarrow} \mathcal{T}'^{\downarrow}, u'^*$. Finally, by Proposition 4, $\mathcal{T}^{\downarrow}, u^* \,\underline{\leftrightarrow}^{\downarrow}\, \mathcal{T}'^{\downarrow}, u'^*$. □

Lemma 14. *Let K be a class of pointed data trees and let Σ be a set of XPath$^{\downarrow}_{=}$-formulas finitely satisfiable in K. Then Σ is satisfiable in some $\downarrow$-quasi ultraproduct of pointed data trees in K.*

Proof. Let $I = \{\Sigma_0 \subset \Sigma \mid \Sigma_0 \text{ is finite}\}$ and for each $\varphi \in \Sigma$, let $\hat{\varphi} = \{i \in I \mid \varphi \in i\}$. Then the set $E = \{\hat{\varphi} \mid \varphi \in \Sigma\}$ has the finite intersection property: $\{\varphi_1, \ldots, \varphi_n\} \in \hat{\varphi_1} \cap \cdots \cap \hat{\varphi_n}$. By the Ultrafilter Theorem (see [4, Prop. 4.1.3]) E can be extended to an ultrafilter U over I.

Since Σ is finitely satisfiable in K, for each $i \in I$ there is $(\mathcal{T}_i, u_i) \in K$ such that $\mathcal{T}_i, u_i \models i$. Let $(\mathcal{T}^{\downarrow}, u^*)$ be the $\downarrow$-quasi ultraproduct of $(\mathcal{T}_i, u_i)_{i\in I}$ modulo U. We show that $\mathcal{T}^{\downarrow}, u^* \models \Sigma$: let $\varphi \in \Sigma$. Then $\hat{\varphi} \in E \subseteq U$ and $\hat{\varphi} \subset \{i \in I \mid \mathcal{T}_i, u_i \models \varphi\}$. Hence $\{i \in I \mid \mathcal{T}_i, u_i \models \varphi\} \in U$, which implies that $\prod_U \mathcal{T}_i \models \mathrm{Tr}_x(\varphi)[u^*]$, where u^* is the ultralimit of $(u_i)_{i\in I}$. Since $\mathcal{T}^{\downarrow} = (\prod_U \mathcal{T}_i)|u^*$, by item 1 of Proposition 8 we conclude that $\mathcal{T}^{\downarrow}, u^* \models \varphi$. □

Theorem 15. *Let K be a class of pointed data trees. Then K is definable by a set of XPath$^{\downarrow}_{=}$-formulas iff K is closed under $\downarrow$-bisimulations and $\downarrow$-quasi ultraproducts, and $\overline{K}$ is closed under $\downarrow$-quasi ultrapowers.*

Proof. For ($\Rightarrow$), suppose that K is definable by a set of XPath$^{\downarrow}_{=}$-formulas. By Theorem 1 it is clear that K is closed under $\downarrow$-bisimulations. By the fundamental theorem of ultraproducts together with item 1 of Proposition 8 it is clear that K is closed under $\downarrow$-quasi ultraproducts. It is also clear that the fundamental theorem of ultraproducts and the fact that any XPath$^{\downarrow}_{=}$-formula is expressible in first-order imply that $\mathcal{T}, u \equiv^{\downarrow} \mathcal{T}^{\downarrow}, u^*$ for any $(\mathcal{T}^{\downarrow}, u^*)$ $\downarrow$-quasi ultrapower modulo U, and therefore that $\overline{K}$ is closed under $\downarrow$-quasi ultrapowers.

For ($\Leftarrow$), suppose K is closed under bisimulations and $\downarrow$-quasi ultraproducts, and $\overline{K}$ is closed under $\downarrow$-quasi ultrapowers. We show that $\Gamma = \bigcap_{(\mathcal{T},u)\in K} \mathrm{Th}_{\downarrow}(\mathcal{T}, u)$ defines K. It is clear that if $(\mathcal{T}, u) \in K$ then $\mathcal{T}, u \models \Gamma$.

Now suppose that $\mathcal{T}, u \models \Gamma$ and consider $\Sigma = \mathrm{Th}_{\downarrow}(\mathcal{T}, u)$. Let Δ be a finite subset of Σ, and assume that Δ is not satisfiable in K. Then $\neg \wedge \Delta$ is true in every pointed data tree of K, so $\neg \wedge \Delta \in \Gamma$. Therefore $\mathcal{T}, u \models \neg \wedge \Delta$ which is a contradiction because $\Delta \subseteq \Sigma$. This shows that Σ is finitely satisfiable in K.

By Lemma 14, Σ is satisfiable in some $\downarrow$-quasi ultraproduct of pointed data trees in K, and since K is closed under $\downarrow$-quasi ultraproducts, Σ is satisfiable in K. Then there exists $(\mathcal{T}', u') \in K$ such that $\mathcal{T}', u' \models \Sigma$ and therefore $\mathcal{T}, u \equiv^{\downarrow} \mathcal{T}', u'$. By Lemma 13, there exist $\downarrow$-quasi ultrapowers $(\mathcal{T}^{\downarrow}, u^*)$ and $(\mathcal{T}'^{\downarrow}, u'^*)$ of $(\mathcal{T}, u)$ and $(\mathcal{T}', u')$ respectively such that $(\mathcal{T}^{\downarrow}, u^*) \underline{\leftrightarrow}^{\downarrow} (\mathcal{T}'^{\downarrow}, u'^*)$. Since K is closed under $\downarrow$-bisimulations, $(\mathcal{T}^{\downarrow}, u^*) \in K$. Suppose $(\mathcal{T}, u) \in \overline{K}$. Since K is closed under $\downarrow$-quasi ultrapowers, $(\mathcal{T}^{\downarrow}, u^*) \in \overline{K}$, and this is a contradiction. Hence we conclude $(\mathcal{T}, u) \in K$. □

Theorem 16. *Let K be a class of pointed data trees. Then K is definable by an XPath$^{\downarrow}_{=}$-formula iff both K and $\overline{K}$ are closed under $\downarrow$-bisimulations and $\downarrow$-quasi ultraproducts.*

Proof. For ($\Rightarrow$) suppose that K is definable by an XPath$^{\downarrow}_{=}$-formula. By Theorem 1 it is clear that K and $\overline{K}$ are closed under bisimulations. By the fundamental theorem of ultraproducts together with item 1 of Proposition 8 it is clear that K and $\overline{K}$ are closed under $\downarrow$-quasi ultraproducts.

For ($\Leftarrow$) suppose K and $\overline{K}$ are closed under bisimulations and $\downarrow$-quasi ultraproducts. Then, by Theorem 15, there exist sets Γ_1 and Γ_2 of XPath$^{\downarrow}_{=}$-formulas defining K and $\overline{K}$ respectively. Consider the set of XPath$^{\downarrow}_{=}$-formulas $\Gamma_1 \cup \Gamma_2$. This set is clearly inconsistent and so, by compactness, there are finite sets Δ_1 and Δ_2 such that $\Delta_i \subseteq \Gamma_i$ $(i = 1, 2)$ and

$$\mathcal{T}, u \models \wedge\Delta_1 \rightarrow \neg \wedge \Delta_2 \tag{1}$$

for any pointed data tree $(\mathcal{T}, u)$. We show that $\varphi = \wedge\Delta_1$ defines K. On the one hand, it is clear that if $(\mathcal{T}, u) \in K$ then $\mathcal{T}, u \models \varphi$. On the other hand, suppose that $\mathcal{T}, u \models \varphi$. From (1) we conclude $\mathcal{T}, u \models \neg \wedge \Delta_2$ and so $\mathcal{T}, u \not\models \Gamma_2$. Then $(\mathcal{T}, u) \notin \overline{K}$ as we wanted to prove. □

In [8, §3.1.1] a restricted version of $\downarrow$-bisimulations, called ℓ-bisimulation, is introduced. It is shown to coincide with the notion of ℓ-equivalence, which informally means *indistinguishable by XPath$_=^\downarrow$ formulas that cannot "see" beyond ℓ 'child'-steps from the current point of evaluation*. Like Theorem 16, the following result characterizes when a class of pointed data trees is definable by a single XPath$_=^\downarrow$-formula. However, instead of using the rather abstract notion of $\downarrow$-quasi ultraproducts, it uses the perhaps more natural notion of ℓ-bisimulation.

Theorem 17. *Let K be a class of pointed data trees. Then K is definable by a formula of XPath$_=^\downarrow$ iff K is closed by ℓ-bisimulations for XPath$_=^\downarrow$ for some ℓ.*

Proof. ($\Rightarrow$) is a direct consequence of Theorem 1. Let us see ($\Leftarrow$). We know [8, Cor. 3.2] that $\{\mathcal{T}', u' \mid \mathcal{T}, u \equiv_\ell^\downarrow \mathcal{T}', u'\}$ is definable by an XPath$_=^\downarrow$-formula $\chi_{\ell,\mathcal{T},u}$ of downward depth $\leq \ell$. We show that

$$\varphi = \bigvee_{(\mathcal{T},u)\in K} \chi_{\ell,\mathcal{T},u}$$

defines K. In [8, Prop. 3.1] it is shown that $\equiv_\ell^\downarrow$ has finite index, and therefore the above disjunction is equivalent to a finite one. On the one hand, if $\mathcal{T}', u' \in K$ then it is clear that $\mathcal{T}', u' \models \chi_{\ell,\mathcal{T}',u'}$ and so $\mathcal{T}', u' \models \varphi$. On the other hand, we have $\mathcal{T}', u' \models \varphi$ iff there is $(\mathcal{T}, u) \in K$ such that $\mathcal{T}', u' \models \chi_{\ell,\mathcal{T},u}$ iff there is $(\mathcal{T}, u) \in K$ such that $\mathcal{T}, u \mathrel{\underline{\leftrightarrow}}_\ell^\downarrow \mathcal{T}', u'$. Hence since K is closed under $\underline{\leftrightarrow}_\ell^\downarrow$, if $\mathcal{T}', u' \models \varphi$ we have $\mathcal{T}', u' \in K$. □

We turn to the vertical fragment.

Lemma 18. *Let $(\mathcal{T}, u)$ and $(\mathcal{T}', u')$ be two pointed data trees such that $\mathcal{T}, u \equiv^\updownarrow \mathcal{T}', u'$. Then there exist $\updownarrow$-quasi ultrapowers $(\mathcal{T}^\updownarrow, u^*)$ and $(\mathcal{T}'^\updownarrow, u'^*)$ of $(\mathcal{T}, u)$ and $(\mathcal{T}', u')$ respectively such that $(\mathcal{T}^\updownarrow, u^*) \mathrel{\underline{\leftrightarrow}}^\updownarrow (\mathcal{T}'^\updownarrow, u'^*)$*

Proof. The proof is analogous to the proof of Lemma 13 but using item 2 instead of item 1 of Proposition 9 and Proposition 6 instead of Proposition 4. □

Lemma 19. *Let K be a class of k-bounded pointed data trees and let Σ be a set of XPath$_=^\updownarrow$-formulas finitely satisfiable in K. Then Σ is satisfiable in some $\updownarrow$-quasi ultraproduct of pointed data trees in K.*

Proof. The proof is analogous to the proof of Lemma 14 but taking $\updownarrow$-quasi ultraproducts instead of $\downarrow$-quasi ultraproducts and using item 2 instead of item 1 of Proposition 8. To apply this Proposition, one has to note that $u^* \in \mathcal{T}^\updownarrow$ since the $\mathcal{T}_i, u_i$ are k-bounded pointed. □

In the next two theorems, the universe of pointed data trees is restricted to those which are k-bounded (for any fixed k). Therefore, the operations of closure and complement must be taken with respect to this universe.

Theorem 20. *Over k-bounded pointed data trees: K is definable by a set of XPath$_=^\updownarrow$-formulas iff K is closed under $\updownarrow$-bisimulations and $\updownarrow$-quasi ultraproducts, and $\overline{K}$ is closed under $\updownarrow$-quasi ultrapowers.*

Proof. The proof is analogous to the proof of Theorem 15 but replacing pointed data trees for k-bounded pointed data trees and every occurrence of $\downarrow$ for $\updownarrow$. Also, for ($\Rightarrow$), one has to use item 2 instead of item 1 of Proposition 8 and for ($\Leftarrow$), Lemmas 19 and 18 instead of Lemmas 14 and 13. □

Theorem 21. *Over k-bounded pointed data trees: K is definable by an XPath$^{\updownarrow}_{=}$-formula iff both K and $\overline{K}$ are closed under $\updownarrow$-bisimulations and $\updownarrow$-quasi ultraproducts.*

As in Theorem 17, one can also restate Theorem 21 in terms of (r, s, k)-bisimulations for XPath$^{\updownarrow}_{=}$ (see [8, §3.2.3] for a definition).

Theorem 22. *Let K be a class of pointed data trees. Then K is definable by a formula of XPath$^{\updownarrow}_{=}$ iff K is closed by (r, s, k)-bisimulations for XPath$^{\updownarrow}_{=}$ for some r, s, k.*

6 Future Research and Applications

In this work we introduced new tools for showing definability results for the downward and vertical fragments of XPath with (in)equality tests over data trees. The general road to prove these theorems themselves is somewhat similar to the one used for the basic modal logic BML (namely, a detour to first-order), but the new concepts (and their interactions) needed to be used in the context of XPath$_=$ are more sophisticated. The notions of $\downarrow$-saturation and $\updownarrow$-saturation are more refined than the usual notions of BML, as they need to take care of the (in)equality tests over the data. Another difference with respect to the models of BML, namely Krike models, is that models of XPath$_=$ are trees (in particular, connected) and so they are not closed under ultraproducts. Thus the notions of $\downarrow$-quasi and $\updownarrow$-quasi ultraproducts arise. These are variants of the classical first-order ultraproducts, and they are, of course, absent in the BML framework.

Our development may be useful for showing other basic model theoretical results such as separation or interpolation of XPath$^{\downarrow}_{=}$ and XPath$^{\updownarrow}_{=}$. We plan to study those and other properties using the tools introduced in this work.

An interesting question is what can be said about other fragments of XPath$_=$ such as XPath$^{\downarrow\downarrow_*}_{=}$ (‘child’ and ‘descendant’ axes) or XPath$^{\updownarrow\updownarrow^*}_{=}$ (‘child’, ‘parent’, ‘descendant’ and ‘ancestor’ axes). As it is mentioned in [8, §5], the bisimulation notions of these two fragments correspond to those for XPath$^{\downarrow}_{=}$ and XPath$^{\updownarrow}_{=}$ respectively. However, in the case of XPath$^{\downarrow\downarrow_*}_{=}$ and XPath$^{\updownarrow\updownarrow^*}_{=}$, the connection to first-order logic is not clear, as we cannot express *transitive closure*.

We finish with some applications:

Example 23. Let K be the class of pointed data trees $(\mathcal{T}, u)$ where u is the root of $\mathcal{T}$ and $\mathcal{T}$ has some node labeled a. On the one hand, K is definable by a first-order σ-sentence. On the other, K is closed under XPath$^{\updownarrow}_{=}$-bisimulations but not closed under $\updownarrow$-quasi ultraproducts: for $i \in \mathbb{N}$ define $\mathcal{T}_i$ as any tree of height i whose only node labeled a is at distance i from the root, and define u_i

as the root of $\mathcal{T}_i$. By an argument similar to the one used in Example 10 one can show that if $(\mathcal{T}^{\updownarrow}, u^*)$ is any $\updownarrow$-quasi ultraproduct of $(\mathcal{T}_i, u_i)_{i\in\mathbb{N}}$ then no node of $\mathcal{T}^{\updownarrow}$ has label a. By Theorem 20, K is not definable by a set of $\mathrm{XPath}^{\updownarrow}_{=}$-formulas.

Example 24. Let $dist_3(x)$ be the property stating that there are nodes y, z so that $x{\rightarrow}y{\rightarrow}z$ and x, y, z have pairwise distinct data values. It can be checked that the $\mathrm{XPath}^{\updownarrow}_{=}$-formula φ_4 from Figure 1 expresses $dist_3(x)$. Let K be the class of pointed data trees $(\mathcal{T}, u)$, where u is the root of $\mathcal{T}$, and for all $v \in T$ we have $dist_3(v)$. On the one hand, K is definable by the set of $\mathrm{XPath}^{\updownarrow}_{=}$-formulas $\{\neg\langle\downarrow^n [\neg\varphi_4]\rangle \mid n \geq 0\}$. On the other, for $i \in \mathbb{N}$, let $(\mathcal{T}_i, u_i)$ be any pointed data tree not in K, of height at least $i + 1$, where u_i is the root of $\mathcal{T}_i$, and such that for all $v \in T_i$ at distance at most i from u_i we have $dist_3(v)$. Let $(\mathcal{T}^{\updownarrow}, u^*)$ be any $\updownarrow$-quasi ultraproduct of $(\mathcal{T}_i, u_i)_{i\in\mathbb{N}}$. One can see that all nodes of $v \in T^{\updownarrow}$ satisfy $dist_3(v)$, and so $(\mathcal{T}^{\updownarrow}, u^*) \in K$. Therefore $\overline{K}$ is not closed under $\updownarrow$-quasi ultraproducts and by Theorem 21, K is not definable by an $\mathrm{XPath}^{\updownarrow}_{=}$-formula. The reader can verify that K is not closed under $\downarrow$-bisimulations (see [8, Prop. 7.5]) and hence, by Theorem 15, K is not definable by a set of $\mathrm{XPath}^{\downarrow}_{=}$-formulas.

References

1. C. Areces, F. Carreiro, and S. Figueira. Characterization, definability and separation via saturated models. *Theoretical Computer Science*, 2014. To appear.
2. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001.
3. M. Bojańczyk, A. Muscholl, T. Schwentick, and L. Segoufin. Two-variable logic on data trees and XML reasoning. *Journal of the ACM*, 56(3):1–48, 2009.
4. C.C. Chang and H.J. Keisler. *Model theory.* Studies in logic and the foundations of mathematics. North-Holland, 1990.
5. J. Clark and S. DeRose. XML path language (XPath). Website, 1999. W3C Recommendation. `http://www.w3.org/TR/xpath`.
6. M. De Rijke. Modal model theory. *Annals of Pure and Applied Logic*, 1995.
7. M. De Rijke and H. Sturm. Global definability in basic modal logic. *Essays on non-classical logic*, 1:111, 2001.
8. D. Figueira, S. Figueira, and C. Areces. Basic model theory of XPath on data trees. In *ICDT*, pages 50–60, 2014.
9. G. Gottlob, C. Koch, and R. Pichler. Efficient algorithms for processing XPath queries. *ACM Transactions on Database Systems*, 30(2):444–491, 2005.
10. M. Gyssens, J. Paredaens, D. Van Gucht, and G.H.L. Fletcher. Structural characterizations of the semantics of XPath as navigation tool on a document. In *PODS*, pages 318–327. ACM, 2006.
11. N. Kurtonina and M. de Rijke. Bisimulations for temporal logic. *Journal of Logic, Language and Information*, 6:403–425, 1997.
12. N. Kurtonina and M. de Rijke. Simulating without negation. *Journal of Logic and Computation*, 7:503–524, 1997.
13. M. Marx and B. de Rijke. Semantic characterizations of navigational XPath. *SIGMOD Record*, 34(2):41–46, 2005.
14. B. ten Cate. The expressivity of XPath with transitive closure. In Stijn Vansummeren, editor, *PODS*, pages 328–337. ACM, 2006.