Basic Model Theory of XPath on Data Trees

Diego Figueira, Santiago Figueira, Carlos Areces

Basic Model Theory of XPath on Data Trees*

Diego Figueira
University of Edinburgh
UK

Santiago Figueira
Universidad de Buenos Aires
and CONICET
Argentina

Carlos Areces
Universidad Nacional de Córdoba
and CONICET
Argentina

ABSTRACT

We investigate model theoretic properties of XPath with data (in)equality tests over the class of data trees, i.e., the class of trees where each node contains a label from a finite alphabet and a data value from an infinite domain.

We provide notions of (bi)simulations for XPath logics containing the `child`, `descendant`, `parent` and `ancestor` axes to navigate the tree. We show that these notions precisely characterize the equivalence relation associated with each logic. We study formula complexity measures consisting of the number of nested axes and nested subformulas in a formula; these notions are akin to the notion of quantifier rank in first-order logic. We show characterization results for fine grained notions of equivalence and (bi)simulation that take into account these complexity measures. We also prove that positive fragments of these logics correspond to the formulas preserved under (non-symmetric) simulations. We show that the logic including the `child` axis is equivalent to the fragment of first-order logic invariant under the corresponding notion of bisimulation. If upward navigation is allowed the characterization fails but a weaker result can still be established. These results hold over the class of possibly infinite data trees and over the class of finite data trees.

Besides their intrinsic theoretical value, we argue that bisimulations are useful tools to prove (non)expressivity results for the logics studied here, and we substantiate this claim with examples.

Categories and Subject Descriptors

F.4.1 [**Mathematical Logic**]: Model theory; H.2.3 [**Languages**]: Query Languages; I.7.2 [**Document Preparation**]: Markup Languages

*This work was partially supported by grant ANPCyT-PICT-2010-688, ANPCyT-PICT-2011-0365, UBACyT 20020110100025 and the FP7-PEOPLE-2011-IRSES Project "Mobility between Europe and Argentina applying Logics to Systems" (MEALS) and the Laboratoire International Associé "INFINIS".

General Terms

Theory, Languages

1. INTRODUCTION

We study the expressive power and model theory of XPath—arguably the most widely used XML query language. Indeed, XPath is implemented in XSLT and XQuery and it is used as a constituent part of 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 [6].

Core-XPath (term coined in [13]) 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 only allows to reason about trees over a finite alphabet. Core-XPath has been well studied and its satisfiability problem is known to be decidable even in the presence of DTDs [17, 1]. Moreover, it is known that it is equivalent to FO^2 (first-order logic with two variables over an appropriate signature on trees) in terms of expressive power [18], and that it is strictly less expressive than PDL with converse over trees [2]. From a database perspective, however, Core-XPath fails to include the single 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 [4]. Here, we will call this logic $\text{XPath}_{=}$. Models of $\text{XPath}_{=}$ are *data trees* which can be seen as XML documents. A data tree is a tree whose nodes contains 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 $\text{XPath}_{=}$ is to allow formulas of the form $\langle \alpha = \beta \rangle$, 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 is true at a node x of a data tree if there are nodes y, z that can be reached by the relations denoted by α, β, respectively, and such that the data value of y is equal to the data value of z.

Recent articles investigate several algorithmic problems

of logics evaluated over data trees. For example, satisfiability and evaluation are discussed in [8, 5]. In particular, all the logics studied in this article have a decidable satisfiability problem [10, 9]; but tools to investigate their *expressive power* are still lacking. There are good reasons for this: in the presence of joins and data values, classical notions such as Ehrenfeucht-Fraïssé games or structural bisimulations are difficult to handle. In this article we take the first steps towards understanding the expressive power and model theory of XPath$_=$ on data trees.

Contribution: XPath$_=$ can navigate the data tree by means of its axes: `child` (that we will note $\downarrow$), `descendant` ($\downarrow_*$), `parent` ($\uparrow$), `ancestor` ($\uparrow^*$), etc. XPath$_=$ can also navigate the data tree horizontally, by going to a next or previous `sibling` of the current node. However, we focus on the vertical axes that allow downward and upward exploration. In particular, we will discuss the following languages: XPath$_=^{\downarrow}$ (XPath$_=$ with $\downarrow$); XPath$_=^{\updownarrow}$ (XPath$_=$ with $\downarrow$ and $\uparrow$); XPath$_=^{\downarrow\downarrow^*}$ (XPath$_=$ with $\downarrow$ and $\downarrow_*$); XPath$_=^{\updownarrow\updownarrow^*}$ (XPath$_=$ with $\downarrow$, $\uparrow$, $\downarrow_*$ and $\uparrow^*$); and its positive fragments. Our main contributions can be summarized as follows:

• In §3 and §5 we introduce bisimulation notions for XPath$_=^{\downarrow}$, XPath$_=^{\downarrow\downarrow^*}$, XPath$_=^{\updownarrow}$, and XPath$_=^{\updownarrow\updownarrow^*}$ and show that they precisely characterize the logical equivalence relation of the respective logic. We also consider fine grained versions of these bisimulations that take into account the number of nested axes and subformulas. The notion of bisimulation for XPath$_=^{\updownarrow}$ relies on a strong normal form which we also introduce.

• In §4 we show that the simulations associated to the defined bisimulations characterize the positive fragments of the logics: a formula is equivalent to a positive formula if and only if it is invariant under simulations.

• In §6 we characterize XPath$_=^{\downarrow}$ as the fragment of first-order logic over data trees (over a signature that includes the `child` relation and an equivalence relation) that is invariant under bisimulations. If we consider XPath$_=^{\updownarrow}$ instead the characterization fails, but a weaker result can still be established.

• Using bisimulations we show (non)expressivity results about XPath$_=$ in §7. We characterize, for example, in which cases increasing the nesting depth increases the expressive power of XPath$_=^{\downarrow}$.

• All results are proved both over the class of arbitrary (possibly infinite) data trees, and over the class of finite data trees.

Related work: The notion of bisimulation was introduced independently by Van Benthem [26] in the context of modal correspondence theory, Milner [19] and Park [23] in concurrency theory, and Forti and Honsell [11] in non-wellfounded set theory (see [25] for a historical outlook). This classical work defines a *standard notion of bisimulation* but this notion has to be suitably adapted for a particular, given logic. The notion of bisimulation for a given logic $\mathcal{L}$ defines when two models are indistinguishable for $\mathcal{L}$, that is, when there is no formula of $\mathcal{L}$ that is true in one model but false in the other. Bisimulations can also be used to obtain model theoretic characterizations that identifies the expressive power of a logic $\mathcal{L}_1$ in terms of the bisimulation invariant fragment of a logic $\mathcal{L}_2$ which, hopefully, is better understood. The challenge, here, is to pinpoint both the appropriate notion of bisimulation required and the adequate 'framework' logic $\mathcal{L}_2$. The classical example of a result of this kind is Van Benthem's characterization for the basic modal logic as the bisimulation (with the standard notion of bisimulation) invariant fragment of first-order logic [26]. Van Benthem's original result over arbitrary structures was proved to hold for finite structures by Rosen [24]. The proof was then simplified and unified by Otto [20, 22], and later expanded by Dawar and Otto [7] to other classes of structures.

Logics for semi-structured databases can often be seen as modal logics. In fact, structural characterizations for XPath without equality test were studied in [14], and XPath is known to be captured by PDL [15], whose bisimulation is well-understood [3]. It is then natural to look for an intuitive bisimulation definition for XPath$_=$.

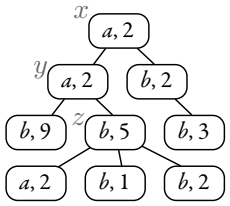

Figure 1: A data tree of *Trees*($\mathbb{A}\times\mathbb{D}$) with $\mathbb{A} = \{a, b\}$ and $\mathbb{D} = \mathbb{N}$.

2. PRELIMINARIES

2.1 Notation

Let $\mathbb{N} = \{1, 2, 3, \dots\}$ and let $[n] := \{1, \dots, n\}$ for $n \in \mathbb{N}$. We use the symbol $\mathbb{A}$ to denote a finite alphabet, and $\mathbb{D}$ to denote an infinite domain (e.g., $\mathbb{N}$) of **data values**. In our examples we will consider $\mathbb{D} = \mathbb{N}$. We write $X{\sim}Y$ to say that X is the result of replacing every data value $d \in \mathbb{D}$ from Y by $f(d)$ where $f : \mathbb{D} \to \mathbb{D}$ is some arbitrary bijection, for any objects X, Y. We write λ for the empty string.

2.2 Data trees

Let *Trees*(A) be the set of ordered and unranked 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**. Figure 1 shows an example of a (finite) data tree. 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, v, w as variables for nodes. Given a node $x \in T$ of $\mathcal{T}$, we write $label(x) \in \mathbb{A}$ to denote the node's label, and $data(x) \in \mathbb{D}$ 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. $(x\overset{n}{\to})$ denotes the set of all descendants of x at distance n, and $(\overset{n}{\to}y)$ denotes the sole ancestor of y at distance n (assuming it has one).

For any binary relation R over elements of data trees, we say that a property P is **R-invariant** whenever the following condition holds: for every data tree $\mathcal{T}$ and $u \in T$, if $(\mathcal{T}, u)$ satisfies P and $(\mathcal{T}, u)$ is R-related to $(\mathcal{T}', u')$ then $(\mathcal{T}', u')$ satisfies P.

2.3 XPath

$$
\begin{array}{rlrl}
[\![\downarrow]\!]^{\mathcal{T}} &= \{(x,y) \mid x{\rightarrow}y\} & [\![\downarrow_*]\!]^{\mathcal{T}} &= \text{reflexive transitive closure of } [\![\downarrow]\!]^{\mathcal{T}} \\
[\![\uparrow]\!]^{\mathcal{T}} &= \{(x,y) \mid y{\rightarrow}x\} & [\![\uparrow^*]\!]^{\mathcal{T}} &= \text{reflexive transitive closure of } [\![\uparrow]\!]^{\mathcal{T}} \\
[\![\varepsilon]\!]^{\mathcal{T}} &= \{(x,x) \mid x \in T\} & [\![a]\!]^{\mathcal{T}} &= \{x \in T \mid \mathit{label}(x) = a\} \\
[\![[\varphi]]\!]^{\mathcal{T}} &= \{(x,x) \mid x \in [\![\varphi]\!]^{\mathcal{T}}\} & [\![\alpha\beta]\!]^{\mathcal{T}} &= \{(x,z) \mid (\exists y \in T)\ (x,y) \in [\![\alpha]\!]^{\mathcal{T}}, (y,z) \in [\![\beta]\!]^{\mathcal{T}}\} \\
[\![\neg\varphi]\!]^{\mathcal{T}} &= T \setminus [\![\varphi]\!]^{\mathcal{T}} & [\![\langle\alpha\rangle]\!]^{\mathcal{T}} &= \{x \in T \mid (\exists y \in T)\ (x,y) \in [\![\alpha]\!]^{\mathcal{T}}\} \\
[\![\alpha \cup \beta]\!]^{\mathcal{T}} &= [\![\alpha]\!]^{\mathcal{T}} \cup [\![\beta]\!]^{\mathcal{T}} & [\![\langle\alpha = \beta\rangle]\!]^{\mathcal{T}} &= \{x \in T \mid (\exists y,z \in T)(x,y) \in [\![\alpha]\!]^{\mathcal{T}}, (x,z) \in [\![\beta]\!]^{\mathcal{T}}, \mathit{data}(y) = \mathit{data}(z)\} \\
[\![\varphi \wedge \psi]\!]^{\mathcal{T}} &= [\![\varphi]\!]^{\mathcal{T}} \cap [\![\psi]\!]^{\mathcal{T}} & [\![\langle\alpha \neq \beta\rangle]\!]^{\mathcal{T}} &= \{x \in T \mid (\exists y,z \in T)(x,y) \in [\![\alpha]\!]^{\mathcal{T}}, (x,z) \in [\![\beta]\!]^{\mathcal{T}}, \mathit{data}(y) \neq \mathit{data}(z)\}
\end{array}
$$

Table 1: Semantics of XPath$_=$ for a data tree $\mathcal{T}$.

We introduce the query language XPath adapted to data trees as abstractions of XML documents. 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. 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 fragment XPath$_=(\mathcal{O})$, with $\mathcal{O} \subseteq \{\downarrow, \downarrow_*, \uparrow, \uparrow^*\}$, is defined by mutual recursion as follows:

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

A **formula** of XPath$_=(\mathcal{O})$ is either a node expression or a path expression. To save space, we use XPath$^{\downarrow}_{=}$ for XPath$_=(\downarrow)$; XPath$^{\updownarrow}_{=}$ for XPath$_=(\downarrow, \uparrow)$; XPath$^{\downarrow\downarrow_*}_{=}$ for XPath$_=(\downarrow, \downarrow_*)$; and XPath$^{\updownarrow\downarrow^*}_{=}$ for XPath$_=(\downarrow, \uparrow, \downarrow_*, \uparrow^*)$.

We formally define the semantics of XPath$_=$ in Table 1. As an example, if $\mathcal{T}$ is the data tree shown in Figure 1, then $[\![\langle\downarrow_*[b \wedge \langle\downarrow[b] \neq \downarrow[b]\rangle]\rangle]\!]^{\mathcal{T}} = \{x, y, z\}$, where the formula reads: "*there is a descendant node labeled b, with two children labeled b with different data values.*" 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. We say that the formulas φ, ψ of XPath$_=$ are **equivalent** (notation: $\varphi \equiv \psi$) iff $[\![\varphi]\!]^{\mathcal{T}} = [\![\psi]\!]^{\mathcal{T}}$ for all data trees $\mathcal{T}$. Similarly, path expressions α, β of XPath$_=$ are **equivalent** (notation: $\alpha \equiv \beta$) iff $[\![\alpha]\!]^{\mathcal{T}} = [\![\beta]\!]^{\mathcal{T}}$ for all data trees $\mathcal{T}$.

We call **downward XPath** to XPath$^{\downarrow}_{=}$ and **vertical XPath** to XPath$^{\updownarrow}_{=}$.

In terms of expressive power, it is easy to see that $\cup$ is unessential: every XPath$_=$ node expression φ has an equivalent φ' with no $\cup$ in its path expressions. φ' can be computed in exponential time without incrementing the number of nested axes or the number of nested subformulas. It is enough to use the following equivalences to eliminate occurrences of $\cup$

$$
\begin{array}{c}
\langle\alpha \odot \beta\rangle \equiv \langle\beta \odot \alpha\rangle \\
\langle\beta(\alpha \cup \alpha')\beta'\rangle \equiv \langle\beta\alpha\beta'\rangle \vee \langle\beta\alpha'\beta'\rangle \\
\langle\gamma \odot \beta(\alpha \cup \alpha')\beta'\rangle \equiv \langle\gamma \odot \beta\alpha\beta'\rangle \vee \langle\gamma \odot \beta\alpha'\beta'\rangle
\end{array}
$$

where $\odot \in \{=, \neq\}$. We will henceforth assume that formulas do not contain union of path expressions.

3. BISIMULATION

3.1 Downward XPath

We write $\mathrm{dd}(\varphi)$ to denote the **downward depth** of φ, defined in Table 2. Let ℓ-XPath$^{\downarrow}_{=}$ be the fragment of XPath$^{\downarrow}_{=}$ consisting of all formulas φ with $\mathrm{dd}(\varphi) \leq \ell$.

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 XPath$^{\downarrow}_{=}$** (notation: $\mathcal{T}, u \equiv^{\downarrow} \mathcal{T}', u'$) iff for all formulas $\varphi \in$ XPath$^{\downarrow}_{=}$, we have $\mathcal{T}, u \models \varphi$ iff $\mathcal{T}', u' \models \varphi$. We say that $\mathcal{T}, u$ and $\mathcal{T}', u'$ are **ℓ-equivalent for XPath$^{\downarrow}_{=}$** (notation: $\mathcal{T}, u \equiv^{\downarrow}_{\ell} \mathcal{T}', u'$) iff for all $\varphi \in \ell$-XPath$^{\downarrow}_{=}$, we have $\mathcal{T}, u \models \varphi$ iff $\mathcal{T}', u' \models \varphi$.

For every ℓ, there are finitely many different formulas φ of $\mathrm{dd}(\varphi) \leq \ell$ up to logical equivalence.

PROPOSITION 3.1. *$\equiv^{\downarrow}_{\ell}$ has finite index.*

COROLLARY 3.2. *$\{\mathcal{T}', u' \mid \mathcal{T}, u \equiv^{\downarrow}_{\ell} \mathcal{T}', u'\}$ is definable by an ℓ-XPath$^{\downarrow}_{=}$-formula $\chi_{\ell,\mathcal{T},u}$.*

3.1.1 Bisimulation and ℓ-bisimulation

Let $\mathcal{T}$ and $\mathcal{T}'$ be two data-trees. We say that $u \in T$ and $u' \in T'$ are **bisimilar for XPath$^{\downarrow}_{=}$** (notation: $\mathcal{T}, u \leftrightarrows^{\downarrow} \mathcal{T}', u'$) 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 (Figure 2):** If xZx', $x \xrightarrow{n} v$ and $x \xrightarrow{m} w$ then there are $v', w' \in T'$ such that $x' \xrightarrow{n} v'$, $x' \xrightarrow{m} w'$ and
  1. $\mathit{data}(v) = \mathit{data}(w) \Leftrightarrow \mathit{data}(v') = \mathit{data}(w')$,
  2. $(\xrightarrow{i} v)\, Z\, (\xrightarrow{i} v')$ for all $0 \leq i < n$, and
  3. $(\xrightarrow{i} w)\, Z\, (\xrightarrow{i} w')$ for all $0 \leq i < m$.
- **Zag:** If xZx', $x' \xrightarrow{n} v'$ and $x' \xrightarrow{m} w'$ then there are $v, w \in T$ such that $x \xrightarrow{n} v$, $x \xrightarrow{m} w$ and items 1, 2 and 3 above are verified.

For a data tree $\mathcal{T}$ and $u \in T$, let $\mathcal{T}|u$ denote the subtree of $\mathcal{T}$ induced by $\{v \in T \mid (\exists n)\ u \xrightarrow{n} v\}$. Observe that the root of $\mathcal{T}|u$ is u. The following results are straightforward consequences of the definition of bisimulation:

PROPOSITION 3.3. $\mathcal{T}, u \leftrightarrows^{\downarrow} (\mathcal{T}|u), u$.

PROPOSITION 3.4. *If $\mathcal{T}$ is a subtree of $\mathcal{T}'$ and $u \in T$ then $\mathcal{T}, u \leftrightarrows^{\downarrow} \mathcal{T}', u$.*

We say that $u \in T$ and $u' \in T'$ are **ℓ-bisimilar for XPath$^{\downarrow}_{=}$** (notation: $\mathcal{T}, u \leftrightarrows^{\downarrow}_{\ell} \mathcal{T}', u'$) if there is a family of relations $(Z_j)_{j \leq \ell}$ in $T \times T'$ such that $uZ_\ell u'$ and for all $j \leq \ell$, $x \in T$ and $x' \in T'$ we have

Downward depth	Vertical depth	Nesting depth
$\mathrm{dd}(a) = 0$	$\mathrm{vd}(a) = (0,0)$	$\mathrm{nd}(a) = 0$
$\mathrm{dd}(\varphi \wedge \psi) = \max\{\mathrm{dd}(\varphi), \mathrm{dd}(\psi)\}$	$\mathrm{vd}(\varphi \wedge \psi) = \max\{\mathrm{vd}(\varphi), \mathrm{vd}(\psi)\}$	$\mathrm{nd}(\varphi \wedge \psi) = \max\{\mathrm{nd}(\varphi), \mathrm{nd}(\psi)\}$
$\mathrm{dd}(\neg\varphi) = \mathrm{dd}(\varphi)$	$\mathrm{vd}(\neg\varphi) = \mathrm{vd}(\varphi)$	$\mathrm{nd}(\neg\varphi) = \mathrm{nd}(\varphi)$
$\mathrm{dd}(\langle\alpha\rangle) = \mathrm{dd}(\alpha)$	$\mathrm{vd}(\langle\alpha\rangle) = \mathrm{vd}(\alpha)$	$\mathrm{nd}(\langle\alpha\rangle) = \mathrm{nd}(\alpha)$
$\mathrm{dd}(\langle\alpha \odot \beta\rangle) = \max\{\mathrm{dd}(\alpha), \mathrm{dd}(\beta)\}$	$\mathrm{vd}(\langle\alpha \odot \beta\rangle) = \max\{\mathrm{vd}(\alpha), \mathrm{vd}(\beta)\}$	$\mathrm{nd}(\langle\alpha \odot \beta\rangle) = \max\{\mathrm{nd}(\alpha), \mathrm{nd}(\beta)\}$
$\mathrm{dd}(\lambda) = 0$	$\mathrm{vd}(\lambda) = (0,0)$	$\mathrm{nd}(\alpha\beta) = \max\{\mathrm{nd}(\alpha), \mathrm{nd}(\beta)\}$
$\mathrm{dd}(\varepsilon\alpha) = \mathrm{dd}(\alpha)$	$\mathrm{vd}(\varepsilon\alpha) = \mathrm{vd}(\alpha)$	$\mathrm{nd}(\varepsilon) = 0$
$\mathrm{dd}([\varphi]\alpha) = \max\{\mathrm{dd}(\varphi), \mathrm{dd}(\alpha)\}$	$\mathrm{vd}([\varphi]\alpha) = \max\{\mathrm{vd}(\varphi), \mathrm{vd}(\alpha)\}$	$\mathrm{nd}([\varphi]) = 1 + \mathrm{nd}(\varphi)$
$\mathrm{dd}(\downarrow\alpha) = 1 + \mathrm{dd}(\alpha)$	$\mathrm{vd}(\downarrow\alpha) = \max\{(0,0), \mathrm{vd}(\alpha) + (1,-1)\}$	$\mathrm{nd}(\downarrow) = 0$
	$\mathrm{vd}(\uparrow\alpha) = \max\{(0,0), \mathrm{vd}(\alpha) + (-1,1)\}$	$\mathrm{nd}(\uparrow) = 0$

Table 2: Definitions of downward depth, vertical depth and nesting depth. ($a \in \mathbb{A}$, $\odot \in \{=, \neq\}$, '+' and 'max' are performed component-wise, α is any path expression or the empty string λ.)

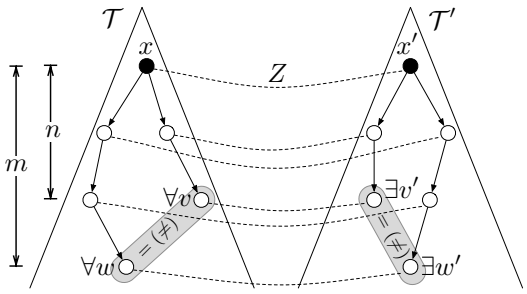

Figure 2: *Zig* clause of bisimulation for XPath$^{\downarrow}_{=}$.

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

Clearly if $\mathcal{T}, u \leftrightarrows^{\downarrow} \mathcal{T}', u'$ then $\mathcal{T}, u \leftrightarrows^{\downarrow}_{\ell} \mathcal{T}', u'$ for all ℓ.

Proposition 3.5. *Suppose $\mathcal{T}$ and $\mathcal{T}'$ have height at most ℓ, $u \in T$, and $u' \in \mathcal{T}'$. Then $\mathcal{T}, u \leftrightarrows^{\downarrow}_{\ell} \mathcal{T}', u'$ iff $\mathcal{T}, u \leftrightarrows^{\downarrow} \mathcal{T}', u'$.*

For a data tree $\mathcal{T}$ and $u \in T$, let $\mathcal{T}|_{\ell} u$ denote the subtree of $\mathcal{T}$ induced by $\{v \in T \mid (\exists n \leq \ell)\ u \xrightarrow{n} v\}$.

Proposition 3.6. $\mathcal{T}, u \leftrightarrows^{\downarrow}_{\ell} (\mathcal{T}|_{\ell} u), u$.

3.1.2 Equivalence and bisimulation

We now show that $\leftrightarrows^{\downarrow}$ coincides with $\equiv^{\downarrow}$ on finitely branching data trees, and that $\leftrightarrows^{\downarrow}_{\ell}$ coincides with $\equiv^{\downarrow}_{\ell}$.

Theorem 3.7.

1. *$\mathcal{T}, u \leftrightarrows^{\downarrow} \mathcal{T}', u'$ implies $\mathcal{T}, u \equiv^{\downarrow} \mathcal{T}', u'$. The converse also holds when $\mathcal{T}$ and $\mathcal{T}'$ are finitely branching.*
2. *$\mathcal{T}, u \leftrightarrows^{\downarrow}_{\ell} \mathcal{T}', u'$ iff $\mathcal{T}, u \equiv^{\downarrow}_{\ell} \mathcal{T}', u'$.*

The Theorem above (see Appendix for details) is a consequence of the next two propositions:

Proposition 3.8. *$\mathcal{T}, u \leftrightarrows^{\downarrow}_{\ell} \mathcal{T}', u'$ implies $\mathcal{T}, u \equiv^{\downarrow}_{\ell} \mathcal{T}', u'$.*

Proof. We actually show that if $\mathcal{T}, u \leftrightarrows^{\downarrow}_{\ell} \mathcal{T}', u'$ via $(Z_i)_{i \leq \ell}$ then for all $0 \leq n \leq j \leq \ell$, for all φ with $\mathrm{dd}(\varphi) \leq j$, and for all α with $\mathrm{dd}(\alpha) \leq j$:

1. If xZ_jx' then $\mathcal{T}, x \models \varphi$ iff $\mathcal{T}', x' \models \varphi$;
2. If $x \xrightarrow{n} v$, $x' \xrightarrow{n} v'$ and $(\xrightarrow{i} v)\, Z_{(j-n)+i}\, (\xrightarrow{i} v')$ for all $0 \leq i \leq n$, then $(x, v) \in [\![\alpha]\!]^{\mathcal{T}}$ iff $(x', v') \in [\![\alpha]\!]^{\mathcal{T}'}$.

We show 1 and 2 by induction on $|\varphi| + |\alpha|$.

Let us see item 1. The base case is $\varphi = a$ for some $a \in \mathbb{A}$. By Harmony, $label(x) = label(x')$ and then $\mathcal{T}, x \models \varphi$ iff $\mathcal{T}', x' \models \varphi$. The Boolean cases for φ are straightforward.

Suppose $\varphi = \langle \alpha = \beta \rangle$. We show $\mathcal{T}, x \models \varphi \Rightarrow \mathcal{T}', x' \models \varphi$, so assume $\mathcal{T}, x \models \varphi$. Suppose there are $v, w \in T$ and $n, m \leq j$ such that $x \xrightarrow{n} v$, $x \xrightarrow{m} w$, $(x, v) \in [\![\alpha]\!]^{\mathcal{T}}$, $(x, w) \in [\![\beta]\!]^{\mathcal{T}}$ and $data(v) = data(w)$. By Zig, there are $v', w' \in T'$ such that $x' \xrightarrow{n} v'$, $x' \xrightarrow{m} w'$, $(\xrightarrow{i} v)\, Z_{j-n+i}\, (\xrightarrow{i} v')$ for all $0 \leq i \leq n$, $(\xrightarrow{i} w)\, Z_{j-m+i}\, (\xrightarrow{i} w')$ for all $0 \leq i \leq m$, and $data(v') = data(w')$. By inductive hypothesis 2 (twice), $(x', v') \in [\![\alpha]\!]^{\mathcal{T}'}$ and $(x', w') \in [\![\beta]\!]^{\mathcal{T}'}$. Hence $\mathcal{T}', x' \models \varphi$. The implication $\mathcal{T}', x' \models \varphi \Rightarrow \mathcal{T}, x \models \varphi$ is analogous. The case $\varphi = \langle \alpha \neq \beta \rangle$ is shown similarly. The case $\varphi = \langle \alpha \rangle$ is similar (and simpler) to the previous case.

Let us now analyze item 2. We only show the 'only if' direction. The base case is when $\alpha \in \{\varepsilon, \downarrow\}$. If $\alpha = \varepsilon$ then $v = x$ and so $n = 0$. Since $v' = x'$, we conclude $(x', v') \in [\![\alpha]\!]^{\mathcal{T}'}$. If $\alpha = \downarrow$ then $x \to v$ in $\mathcal{T}$, and so $n = 1$. Since $x' \to v'$, we have $(x', v') \in [\![\alpha]\!]^{\mathcal{T}'}$.

For the inductive step, let

$$x_0, \ldots, x_n \in T \quad \text{and} \quad x'_0, \ldots, x'_n \in T'$$

be such that

$$x = x_0 \to x_1 \to x_2 \to \cdots \to x_n = v \qquad \text{in } \mathcal{T},$$

$$x' = x'_0 \to x'_1 \to x'_2 \to \cdots \to x'_n = v' \qquad \text{in } \mathcal{T}',$$

and $x_i Z_{j-i} x'_i$ for all $0 \leq i \leq n$. Assume, for contradiction, that $(x', v') \notin [\![\alpha]\!]^{\mathcal{T}'}$. Then, there is a subformula φ of α and $k \in \{0, \ldots, n\}$ such that $\mathcal{T}, x_k \models \varphi$ and $\mathcal{T}', x'_k \not\models \varphi$(this is shown in Lemma A.1 in the Appendix). This contradicts the inductive hypothesis 1. $\square$

PROPOSITION 3.9. *$\mathcal{T}, u \equiv^{\downarrow}_{\ell} \mathcal{T}', u'$ implies $\mathcal{T}, u \leftrightarrows^{\downarrow}_{\ell} \mathcal{T}', u'$.*

PROOF. Fix $u \in T$ and $u' \in T'$ such that $\mathcal{T}, u \equiv^{\downarrow}_{\ell} \mathcal{T}', u'$. Define $(Z_i)_{i \leq \ell}$ by

$$x Z_i x' \quad \text{iff} \quad \mathcal{T}, x \equiv^{\downarrow}_{i} \mathcal{T}', x'.$$

We show that Z is an ℓ-bisimulation between $\mathcal{T}, u$ and $\mathcal{T}', u'$. By hypothesis, $u Z_\ell u'$. Fix $h \leq \ell$, by construction, Z_h satisfies Harmony. Let us see that Z_h satisfies Zig (the case for Zag is analogous). Suppose $x Z_h x'$,

$$x = v_0 \to v_1 \to \cdots \to v_n = v \qquad \text{in } \mathcal{T},$$
$$x = w_0 \to w_1 \to \cdots \to w_m = w \qquad \text{in } \mathcal{T},$$

and $data(v) = data(w)$ (the case $data(v) \neq data(w)$ is shown in a similar way), where $m, n \leq h$. Let $P \subseteq T'^2$ be defined by

$$P = \{(v', w') \mid x' \xrightarrow{n} v' \wedge x' \xrightarrow{m} w' \wedge data(v') = data(w')\}.$$

Since $\mathcal{T}, x \equiv^{\downarrow}_{h} \mathcal{T}', x'$, $\mathrm{dd}(\langle \downarrow^n = \downarrow^m \rangle) \leq h$ and $\mathcal{T}, x \models \langle \downarrow^n = \downarrow^m \rangle$, we conclude that $P \neq \emptyset$. We next show that there exists $(v', w') \in P$ such that

i. $x' = v'_0 \to v'_1 \to \cdots \to v'_n = v'$ in $\mathcal{T}'$,

ii. $x' = w'_0 \to w'_1 \to \cdots \to w'_m = w'$ in $\mathcal{T}'$,

iii. $(\forall i \in \{0, \ldots, n\})$ $\mathcal{T}, v_i \equiv^{\downarrow}_{h-i} \mathcal{T}', v'_i$, and

iv. $(\forall j \in \{0, \ldots, m\})$ $\mathcal{T}, w_j \equiv^{\downarrow}_{h-j} \mathcal{T}', w'_j$,

and hence Zig is satisfied by Z_h. By way of contradiction, assume that for all $(v', w') \in P$ satisfying i and ii we have either

(a) $(\exists i \in \{0, \ldots, n\})$ $\mathcal{T}, v_i \not\equiv^{\downarrow}_{h-i} \mathcal{T}', v'_i$, or

(b) $(\exists j \in \{0, \ldots, m\})$ $\mathcal{T}, w_j \not\equiv^{\downarrow}_{h-j} \mathcal{T}', w'_j$.

Fix $\top$ as any tautology such that $\mathrm{dd}(\top) = 0$. For each $(v', w') \in P$ we define two families of formulas,

$$\varphi^0_{v',w'}, \ldots, \varphi^n_{v',w'} \quad \text{and} \quad \psi^0_{v',w'}, \ldots, \psi^m_{v',w'},$$

satisfying that $\mathrm{dd}(\varphi^i_{v',w'}) \leq h - i$ for all $i \in \{0, \ldots, n\}$ and $\mathrm{dd}(\psi^j_{v',w'}) \leq h - j$ for all $j \in \{0, \ldots, m\}$ as follows:

- Suppose that (a) holds and that i is the smallest number such that $\mathcal{T}, v_i \not\equiv^{\downarrow}_{h-i} \mathcal{T}', v'_i$. Let $\varphi^i_{v',w'}$ be such that $\mathrm{dd}(\varphi^i_{v',w'}) \leq h-i$ and $\mathcal{T}, v_i \models \varphi^i_{v',w'}$ but $\mathcal{T}', v'_i \not\models \varphi^i_{v',w'}$. For $k \in \{0, \ldots, n\} \setminus \{i\}$, let $\varphi^k_{v',w'} = \top$, and for $k \in \{0, \ldots, m\}$, let $\psi^k_{v',w'} = \top$.
- Suppose that (a) does not hold. Then (b) holds. Let j be the smallest number such that $\mathcal{T}, w_j \not\equiv^{\downarrow}_{h-j} \mathcal{T}', w'_j$. Let $\psi^j_{v',w'}$ be such that $\mathrm{dd}(\psi^j_{v',w'}) \leq h-j$ and $\mathcal{T}, w_j \models \psi^j_{v',w'}$ but $\mathcal{T}', w'_j \not\models \psi^j_{v',w'}$. For $k \in \{0, \ldots, m\} \setminus \{j\}$, let $\psi^k_{v',w'} = \top$, and for $k \in \{0, \ldots, n\}$, let $\varphi^k_{v',w'} = \top$.

For each $i \in \{0, \ldots, n\}$ and $j \in \{0, \ldots, m\}$, let

$$\Phi^i = \bigwedge_{(v',w') \in P} \varphi^i_{v',w'} \quad \text{and} \quad \Psi^j = \bigwedge_{(v',w') \in P} \psi^j_{v',w'}. \qquad (1)$$

Since $\mathrm{dd}(\varphi^i_{v',w'}) \leq h - i$, by Proposition 3.1, there are finitely many non-equivalent formulas $\varphi^i_{v',w'}$; the same applies to $\psi^j_{v',w'}$. Hence, both infinite conjunctions in (1) are equivalent to finite ones, and we may assume that Φ^i and Ψ^j are well-formed formulas. Finally, let

$$\alpha = [\Phi^0]\downarrow[\Phi^1]\downarrow \cdots \downarrow[\Phi^n] \text{ and } \beta = [\Psi^0]\downarrow[\Psi^1]\downarrow \cdots \downarrow[\Psi^m].$$

By construction, $\mathrm{dd}(\alpha), \mathrm{dd}(\beta) \leq h$ and so $\mathrm{dd}(\langle \alpha = \beta \rangle) \leq h$. Furthermore, $\mathcal{T}, x \models \langle \alpha = \beta \rangle$ and $\mathcal{T}', x' \not\models \langle \alpha = \beta \rangle$. This contradicts $\mathcal{T}, x \equiv^{\downarrow}_{h} \mathcal{T}', x'$. $\square$

3.2 Vertical XPath

We now study bisimulation for $\mathrm{XPath}^{\updownarrow}_{=}$. Interestingly, the notion we give is simpler than the one for $\mathrm{XPath}^{\downarrow}_{=}$ due to a normal form enjoyed by the logic.

In the downward fragment of $\mathrm{XPath}_{=}$ we used $\mathrm{dd}(\varphi)$ to measure the maximum depth from the current point of evaluation that the formula can access. For the vertical fragment of $\mathrm{XPath}_{=}$, we need to define both the maximum distance r going downward and the maximum distance s going upward that the formula can reach. We call the pair (r, s) the vertical depth of a formula. Formally, the **vertical depth** of a formula φ (notation: $\mathrm{vd}(\varphi)$) is the pair $\mathrm{vd}(\varphi) \in \mathbb{Z}^2_{\geq 0}$ defined in Table 2.

The **nesting depth** of a formula φ (notation: $\mathrm{nd}(\varphi)$) is the maximum number of nested [ ] appearing in φ. See Table 2 for the formal definition.

Let (r, s, k)-$\mathrm{XPath}^{\updownarrow}_{=}$ be the set of all formulas φ in $\mathrm{XPath}^{\updownarrow}_{=}$ with $\mathrm{vd}(\varphi) \leq (r, s)$ and $\mathrm{nd}(\varphi) \leq k$.

Let $\mathcal{T}$ and $\mathcal{T}'$ be data trees, let $u \in T$ and $u' \in T'$. We say that $\mathcal{T}, u$ and $\mathcal{T}', u'$ are **equivalent for $\mathrm{XPath}^{\updownarrow}_{=}$** (notation: $\mathcal{T}, u \equiv^{\updownarrow} \mathcal{T}', u'$) iff for all $\varphi \in \mathrm{XPath}^{\updownarrow}_{=}$, we have $\mathcal{T}, u \models \varphi$ iff $\mathcal{T}', u' \models \varphi$. $\mathcal{T}, x$ and $\mathcal{T}', x'$ are **(r, s)-equivalent** [resp. **(r, s, k)-equivalent**] **for $\mathrm{XPath}^{\updownarrow}_{=}$**, and we note it $\mathcal{T}, x \equiv^{\updownarrow}_{r,s} \mathcal{T}', x'$ [resp. $\mathcal{T}, x \equiv^{\updownarrow}_{r,s,k} \mathcal{T}', x'$] if they satisfy the same $\mathrm{XPath}^{\updownarrow}_{=}$ formulas φ so that $\mathrm{vd}(\varphi) \leq (r, s)$ [resp. $\mathrm{vd}(\varphi) \leq (r, s)$ and $\mathrm{nd}(\varphi) \leq k$].

3.2.1 Normal form

We define a useful normal form for $\mathrm{XPath}^{\updownarrow}_{=}$ that will be implicitly used in the definition of bisimulation in the section. For $n \geq 0$, let $\downarrow^n$ denote the concatenation of n symbols $\downarrow$. I.e., $\downarrow^0$ is the empty string λ, $\downarrow^1 = \downarrow$, and $\downarrow^{n+1} = \downarrow\downarrow^n$ (similarly for $\uparrow^n$).

A path expression α of $\mathrm{XPath}^{\updownarrow}_{=}$ is **downward** [resp. **upward**] if it is of the form $\downarrow^n[\varphi]$ [resp. $[\varphi]\uparrow^n$] for some $n \geq 0$ with $\varphi \in \mathrm{XPath}^{\updownarrow}_{=}$. For example, $\downarrow[\langle\uparrow\rangle]\downarrow$ is a downward expression whereas $\downarrow[\langle\downarrow\rangle]\downarrow$ is not. An **up-down** expression is any expression of the form ε, $\alpha^{\uparrow}$, $\alpha^{\downarrow}$ or $\alpha^{\uparrow}\alpha^{\downarrow}$ where $\alpha^{\uparrow}$ is upward and $\alpha^{\downarrow}$ is downward. Henceforth we will use $\alpha^{\uparrow}, \beta^{\uparrow}, \gamma^{\uparrow}$ to denote upward expressions and $\alpha^{\downarrow}, \beta^{\downarrow}, \gamma^{\downarrow}$ to denote downward expressions and $\alpha^{\updownarrow}, \beta^{\updownarrow}, \gamma^{\updownarrow}$ to denote up-down expressions. Note that in particular any downward or upward expression is an up-down expression. An $\mathrm{XPath}^{\updownarrow}_{=}$ formula or expression is in **up-down normal form** if every path expression contained in it is up-down and every data test is of the form $\langle \varepsilon \odot \alpha^{\updownarrow} \rangle$ with $\odot \in \{=, \neq\}$.

PROPOSITION 3.10. *Let $\varphi \in (r,s,k)$-XPath$^{\updownarrow}_{=}$. There is $\varphi^{\uparrow\downarrow} \in$ XPath$^{\updownarrow}_{=}$ in up-down normal form such that*

1. $\varphi^{\uparrow\downarrow} \equiv \varphi$;
2. $\mathrm{vd}(\varphi^{\uparrow\downarrow}) = (r,s)$; *and*
3. $\mathrm{nd}(\varphi^{\uparrow\downarrow}) \leq k \cdot (r+s+2)$.

3.2.2 Finite index

Contrary to the case of XPath$^{\downarrow}_{=}$ (*cf.*, Proposition 3.1), the logical equivalence relation restricted to XPath$^{\updownarrow}_{=}$-formulas of bounded vertical depth has infinitely many equivalence classes.

PROPOSITION 3.11. *If $r+s \geq 2$ then $\equiv^{\updownarrow}_{r,s}$ has infinite index.*

In the proof of the above proposition (see Appendix)we need to use formulas with unbounded nesting depth. In fact, when restricted to bounded nesting depth there are only finitely many formulas up to logical equivalence, as stated next.

PROPOSITION 3.12. *$\equiv^{\updownarrow}_{r,s,k}$ has finite index.*

COROLLARY 3.13. *$\{\mathcal{T}', u' \mid \mathcal{T}, u \equiv^{\updownarrow}_{r,s,k} \mathcal{T}', u'\}$ is definable by an (r,s,k)-XPath$^{\updownarrow}_{=}$-formula.*

3.2.3 Bisimulation and (r,s,k)-bisimulation

The advantage of the normal form presented in Section 3.2.1, is that it makes it possible to use a very simple notion of bisimulation. The disadvantage is that, since it does not preserve nesting depth, $\leftrightarrows^{\updownarrow}_{r,s,k}$ does not correspond precisely to $\equiv^{\updownarrow}_{r,s,k}$, although $\leftrightarrows^{\updownarrow}$ corresponds precisely to $\equiv^{\updownarrow}$. Nonetheless, we obtain, for all r, s, k,

$$\leftrightarrows_{r,s,k} \subseteq \equiv^{\updownarrow}_{r,s,k} \subseteq \leftrightarrows^{\updownarrow}_{r,s,k\cdot(r+s+2)}.$$

Let $\mathcal{T}$ and $\mathcal{T}'$ be two data-trees. We say that $u \in T$ and $u' \in T'$ are **bisimilar for XPath$^{\updownarrow}_{=}$** (notation: $\mathcal{T}, u \leftrightarrows^{\updownarrow} \mathcal{T}', u'$) 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 $label(x) = label(x')$,
- **Zig (Figure 3):** If xZx', $y \xrightarrow{n} x$ and $y \xrightarrow{m} z$ then there are $y', z' \in T'$ such that $y' \xrightarrow{n} x'$, $y' \xrightarrow{m} z'$, $data(z) = data(x) \Leftrightarrow data(z') = data(x')$, and zZz'.
- **Zag:** If xZx', $y' \xrightarrow{n} x'$ and $y' \xrightarrow{m} z'$ then there are $y, z \in T$ such that $y \xrightarrow{n} x$, $y \xrightarrow{m} z$, $data(z) = data(x) \Leftrightarrow data(z') = data(x')$, and zZz'.

Observe that contrary to the definition of $\leftrightarrows^{\downarrow}$, the conditions above do not require intermediate nodes to be related by Z. This is a direct consequence of the up-down normal form (Proposition 3.10).

We say that $u \in T$ and $u' \in T'$ are **(r,s,k)-bisimilar for XPath$^{\updownarrow}_{=}$** (notation: $\mathcal{T}, u \leftrightarrows^{\updownarrow}_{r,s,k} \mathcal{T}', u'$) if there is a family of relations $(Z^{\hat{k}}_{\hat{r},\hat{s}})_{\hat{r}+\hat{s} \leq r+s, \hat{k} \leq k}$ in $T \times T'$ such that $uZ^{k}_{r,s}u'$ and for all $\hat{r}+\hat{s} \leq r+s, \hat{k} \leq k$, $x \in T$ and $x' \in T'$ we have that the following conditions hold.

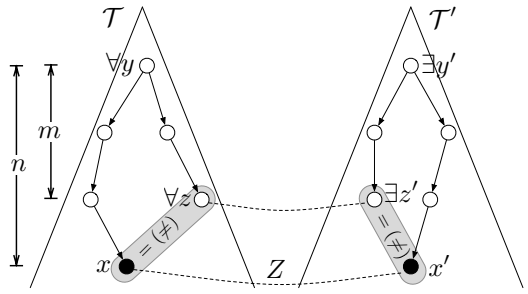

Figure 3: *Zig* clause of bisimulation for XPath$^{\updownarrow}_{=}$

- **Harmony:** If $xZ^{\hat{k}}_{\hat{r},\hat{s}}x'$ then $label(x) = label(x')$.
- **Zig:** If $xZ^{\hat{k}}_{\hat{r},\hat{s}}x'$, $y \xrightarrow{n} x$ and $y \xrightarrow{m} z$ with $n \leq \hat{s}$ and $m \leq \hat{r}+n$ then there are $y', z' \in T'$ such that $y' \xrightarrow{n} x'$, $y' \xrightarrow{m} z'$, and the following hold
  (1) $data(z) = data(x) \Leftrightarrow data(z') = data(x')$,
  (2) if $\hat{k} > 0$, $zZ^{\hat{k}-1}_{\hat{r}',\hat{s}'}z'$ for $\hat{r}' = \hat{r}+n-m$, $\hat{s}' = \hat{s}-n+m$.
- **Zag:** If $xZ^{\hat{k}}_{\hat{r},\hat{s}}x'$, $y' \xrightarrow{n} x'$ and $y' \xrightarrow{m} z'$ with $n \leq \hat{s}$ and $m \leq \hat{r}+n$ then there are $y, z \in T$ such that $y \xrightarrow{n} x$, $y \xrightarrow{m} z$, and items (1) and (2) above are verified.

OBSERVATION 3.14. *If $xZ^{\hat{k}}_{\hat{r},\hat{s}}x'$, $y \xrightarrow{n} x$ and $y' \xrightarrow{n} x'$ then it follows that $yZ^{\hat{k}-1}_{\hat{r}',\hat{s}'}y'$, for $\hat{r}' = \hat{r}+n$, $\hat{s}' = \hat{s}-n$. The same occurs with Z instead of $Z^{\hat{k}}_{\hat{r},\hat{s}}$ for the case of bisimilarity.*

For a data tree $\mathcal{T}$ and $u \in T$, let $\mathcal{T}|^{s}_{r}u$ denote the subtree of $\mathcal{T}$ induced by

$$\{v \in T \mid (\exists m \leq s)\ (\exists n \leq r+m)\ (\exists w \in T)\ w \xrightarrow{m} u \wedge\ w \xrightarrow{n} v\}.$$

PROPOSITION 3.15. *$\mathcal{T}, u \leftrightarrows^{\updownarrow}_{r,s,k} (\mathcal{T}|^{s}_{r}u), u$.*

3.2.4 Equivalence and bisimulation

The next result says that $\leftrightarrows^{\updownarrow}$ coincides with $\equiv^{\updownarrow}$ on finitely branching data trees, and states precisely in what way $\leftrightarrows^{\updownarrow}_{r,s,k}$ is related to $\equiv^{\updownarrow}_{r,s,k}$.

THEOREM 3.16.

1. *$\mathcal{T}, u \leftrightarrows^{\updownarrow} \mathcal{T}', u'$ implies $\mathcal{T}, u \equiv^{\updownarrow} \mathcal{T}', u'$. The converse also holds when $\mathcal{T}$ and $\mathcal{T}'$ are finitely branching.*
2. *$\mathcal{T}, u \leftrightarrows^{\updownarrow}_{r,s,k\cdot(r+s+2)} \mathcal{T}', u'$ implies $\mathcal{T}, u \equiv^{\updownarrow}_{r,s,k} \mathcal{T}', u'$.*
3. *$\mathcal{T}, u \equiv^{\updownarrow}_{r,s,k} \mathcal{T}', u'$ implies $\mathcal{T}, u \leftrightarrows^{\updownarrow}_{r,s,k} \mathcal{T}', u'$.*

COROLLARY 3.17. *$\leftrightarrows^{\updownarrow}_{r,s,k}$ has finite index.*

4. SIMULATION

In this section we define notions of directed (non-symmetric) simulations for XPath$^{\downarrow}_{=}$ and XPath$^{\updownarrow}_{=}$, as it is done, e.g., in [16] for some modal logics. We obtain results similar to Theorems 3.7 and 3.16 but relating each simulation notion with the corresponding logical implication.

We say that an XPath$_=$ formula is **positive** if it contains no negation $\neg$ and no inequality data tests $\langle \alpha \neq \beta \rangle$. For $\mathcal{L}$ one of XPath$^{\downarrow}_{=}$, XPath$^{\updownarrow}_{=}$, XPath$^{\downarrow\downarrow^*}_{=}$, or XPath$^{\updownarrow\updownarrow^*}_{=}$, we write $\mathcal{L}^+$ for the positive fragment of $\mathcal{L}$.

A **simulation for XPath$^{\downarrow}_{=}$** [resp. for XPath$^{\updownarrow}_{=}$] is simply a bisimulation from which the *Zag* clause and half of the first condition in the *Zig* clause have been omitted. Observe that simulations need not be symmetric.

Formally, we say that $u \in T$ is **similar to** $u' \in T'$ **for XPath$^{\downarrow}_{=}$** (notation: $\mathcal{T}, u \rightrightarrows^{\downarrow} \mathcal{T}', u'$) 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 $label(x) = label(x')$.
- **Zig:** If xZx', $x \xrightarrow{n} v$ and $x \xrightarrow{m} w$ then there are $v', w' \in T'$ such that $x' \xrightarrow{n} v'$, $x' \xrightarrow{m} w'$ and
  1. $data(v) = data(w) \Rightarrow data(v') = data(w')$,
  2. $(\xrightarrow{i} v)\, Z\, (\xrightarrow{i} v')$ for all $0 \leq i < n$, and
  3. $(\xrightarrow{i} w)\, Z\, (\xrightarrow{i} w')$ for all $0 \leq i < m$.

$u \in T$ is **similar to** $u' \in T'$ **for XPath$^{\updownarrow}_{=}$** (notation: $\mathcal{T}, u \rightrightarrows^{\updownarrow} \mathcal{T}', u'$) 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 $label(x) = label(x')$.
- **Zig:** If xZx', $y \xrightarrow{n} x$ and $y \xrightarrow{m} z$ then there are $y', z' \in T'$ such that $y' \xrightarrow{n} x'$, $y' \xrightarrow{m} z'$, zZz', and if $data(z) = data(x)$ then $data(z') = data(x')$.

Relations $\rightrightarrows^{\downarrow}_{\ell}$ and $\rightrightarrows^{\updownarrow}_{r,s,k}$ are defined accordingly. We define one-way (non-symmetric) logical implication between models as follows. We write $\mathcal{T}, u \Rrightarrow^{\downarrow} \mathcal{T}', u'$ for

$$(\forall \varphi \in \text{XPath}^{\downarrow+}_{=})\ [\mathcal{T}, u \models \varphi \Rightarrow \mathcal{T}', u' \models \varphi].$$

Define $\Rrightarrow^{\downarrow}_{\ell}$, $\Rrightarrow^{\updownarrow}$, and $\Rrightarrow^{\updownarrow}_{r,s,k}$ in an analogous way for ℓ-XPath$^{\downarrow+}_{=}$, XPath$^{\updownarrow+}_{=}$, (r,s,k)-XPath$^{\updownarrow+}_{=}$, respectively. As for bisimulation, we have that $\rightrightarrows$ coincides with $\Rrightarrow$.

THEOREM 4.1.

1. *Let $\dagger \in \{\downarrow, \updownarrow\}$. $\mathcal{T}, u \rightrightarrows^{\dagger} \mathcal{T}', u'$ implies $\mathcal{T}, u \Rrightarrow^{\dagger} \mathcal{T}', u'$. The converse holds when $\mathcal{T}'$ is finitely branching.*
2. *$\mathcal{T}, u \rightrightarrows^{\downarrow}_{\ell} \mathcal{T}', u'$ iff $\mathcal{T}, u \Rrightarrow^{\downarrow}_{\ell} \mathcal{T}', u'$.*
3. *$\mathcal{T}, u \rightrightarrows^{\updownarrow}_{r,s,k\cdot(r+s+2)} \mathcal{T}', u'$ implies $\mathcal{T}, u \Rrightarrow^{\updownarrow}_{r,s,k} \mathcal{T}', u'$.*
4. *$\mathcal{T}, u \Rrightarrow^{\updownarrow}_{r,s,k} \mathcal{T}', u'$ implies $\mathcal{T}, u \rightrightarrows^{\updownarrow}_{r,s,k} \mathcal{T}', u'$.*

We say that $\mathcal{T}'$ is a **substructure** of $\mathcal{T}$ if $\mathcal{T}'$ is a data tree which results from removing some nodes of $\mathcal{T}$, i.e., $T' \subseteq T$ and for all $u, v \in T'$ we have: 1) $u \rightarrow v$ on $\mathcal{T}$ iff $u \rightarrow v$ on $\mathcal{T}'$; 2) $label(u)$ on $\mathcal{T}'$ equals $label(u)$ on $\mathcal{T}$; and 3) $data(u)$ on $\mathcal{T}'$ equals $data(u)$ on $\mathcal{T}$. Equivalently, seen as σ-structures, $\mathcal{T}'$ is the σ-substructure of $\mathcal{T}$ induced by $T' \subseteq T$. One can verify that the identity on T' is a simulation for XPath$^{\updownarrow}_{=}$ from $\mathcal{T}'$ to $\mathcal{T}$.

LEMMA 4.2. *If $\mathcal{T}'$ is a substructure of $\mathcal{T}$ and $u' \in T'$ then $\mathcal{T}', u' \rightrightarrows^{\updownarrow} \mathcal{T}, u'$.*

We obtain that the formulas of XPath$_=$ invariant under simulations are, precisely, the positive ones.

THEOREM 4.3.

1. *$\varphi \in XPath^{\downarrow}_{=}$ is $\rightrightarrows^{\downarrow}$-invariant [resp. $\rightrightarrows^{\downarrow}_{\ell}$] iff it is equivalent to a formula of $XPath^{\downarrow+}_{=}$ [resp. ℓ-$XPath^{\downarrow+}_{=}$].*
2. *$\varphi \in XPath^{\updownarrow}_{=}$ is $\rightrightarrows^{\updownarrow}$-invariant iff it is equivalent to a formula of $XPath^{\updownarrow+}_{=}$.*
3. *If $\varphi \in XPath^{\updownarrow}_{=}$ is $\rightrightarrows^{\updownarrow}_{r,s,k}$-invariant then it is equivalent to a formula of (r,s,k)-$XPath^{\updownarrow+}_{=}$.*
4. *If $\varphi \in XPath^{\updownarrow}_{=}$ is equivalent to a formula of (r,s,k)-$XPath^{\updownarrow+}_{=}$ then φ is $\rightrightarrows^{\updownarrow}_{r,s,k'}$-invariant, for $k' = k\cdot(r+s+2)$.*

5. ADDING TRANSITIVITY

As it happens, for example, with the basic modal logic and propositional dynamic logic, the same notion of bisimulation [resp. simulation] of each logic captures the logical equivalence [resp. logical implication] for the corresponding fragments including the reflexive-transitive closure of the axes which are present. Intuitively, this occurs because $\downarrow_*$ is an infinite union of compositions of $\downarrow$, and similarly for $\uparrow$.

Let $\equiv^{\downarrow\downarrow^*}$ and $\equiv^{\updownarrow\updownarrow^*}$ be the logical equivalence relation for XPath$^{\downarrow\downarrow^*}_{=}$ and XPath$^{\updownarrow\updownarrow^*}_{=}$ respectively, and let $\Rrightarrow^{\downarrow\downarrow^*}$ and $\Rrightarrow^{\updownarrow\updownarrow^*}$ be the logical implication for XPath$^{\downarrow\downarrow^*+}_{=}$ and XPath$^{\updownarrow\updownarrow^*+}_{=}$ respectively.

THEOREM 5.1. *Let $\dagger \in \{\downarrow\downarrow_*, \updownarrow\updownarrow^*\}$.*

1. *$\mathcal{T}, u \leftrightarrows^{\dagger} \mathcal{T}', u'$ implies $\mathcal{T}, u \equiv^{\dagger} \mathcal{T}', u'$. The converse also holds when $\mathcal{T}'$ is finitely branching.*
2. *$\mathcal{T}, u \rightrightarrows^{\dagger} \mathcal{T}', u'$ implies $\mathcal{T}, u \Rrightarrow^{\dagger} \mathcal{T}', u'$. The converse also holds when $\mathcal{T}'$ is finitely branching.*

6. CHARACTERIZATION

In §6.1 we show that there is a truth-preserving translation from XPath$^{\updownarrow}_{=}$ to first-order logic over an appropriate signature. In §6.2 we characterize XPath$^{\downarrow}_{=}$ as the fragment of first-order logic $\leftrightarrows^{\downarrow}$-invariant over data trees. In §6.3 we show that this result fails for XPath$^{\updownarrow}_{=}$ in general, but a weaker result can still be proved.

6.1 Translating to first-order logic

We say that an XPath$^{\updownarrow}_{=}$-path expression α is in **simple normal form** if it is of the form

$$[\varphi_0] o_1 [\varphi_1] o_2 \cdots o_n [\varphi_n],$$

for $n \geq 0$, $\varphi_i \in \text{XPath}^{\updownarrow}_{=}$, and $o_i \in \{\downarrow, \uparrow\}$.

PROPOSITION 6.1. *For any $XPath^{\updownarrow}_{=}$- [resp. $XPath^{\downarrow}_{=}$-] path expression α there is an equivalent $XPath^{\updownarrow}_{=}$- [resp. $XPath^{\downarrow}_{=}$-] path expression α' in simple normal form. Further, α' can be computed in polynomial time from α.*[1]

We say that an XPath$^{\updownarrow}_{=}$-formula φ is in **simple normal form** if each path expression α occurring in φ is in simple normal form.

Fix the signature σ with binary relations $\rightsquigarrow$ and $\approx$, 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 such that

$$\rightsquigarrow^{\mathcal{T}} \quad = \quad \{(x,y) \in T^2 \mid y \text{ is a child of } x\};$$

[1]Note that this proposition holds only for paths expressions without union.

$$\approx^{\mathcal{T}} = \{(x,y) \in T^2 \mid data(x) = data(y)\};$$
$$P_a^{\mathcal{T}} = \{x \in T \mid label(x) = a\}.$$

We define the following translation Tr mapping $\mathrm{XPath}^{\updownarrow}_{=}$ formulas in simple normal form to first-order σ-formulas:

$$\begin{aligned}
\mathrm{Tr}_x(a) &= P_a(x) && (a \in \mathbb{A})\\
\mathrm{Tr}_x(\varphi \dagger \psi) &= \mathrm{Tr}_x(\varphi) \dagger \mathrm{Tr}_x(\psi) && (\dagger \in \{\wedge, \vee\})\\
\mathrm{Tr}_x(\neg\varphi) &= \neg \mathrm{Tr}_x(\varphi)\\
\mathrm{Tr}_x(\langle\alpha\rangle) &= (\exists \bar{y})\big(x = y_0 \wedge \mathrm{Tr}_{\bar{y}}(\alpha)\big)\\
\mathrm{Tr}_x(\langle\alpha = \beta\rangle) &= (\exists \bar{y})(\exists \bar{z})\big(x = y_0 \wedge x = z_0 \wedge y_n \approx z_m \wedge\\
&\qquad \mathrm{Tr}_{\bar{y}}(\alpha) \wedge \mathrm{Tr}_{\bar{z}}(\beta)\big)\\
\mathrm{Tr}_x(\langle\alpha \neq \beta\rangle) &= (\exists \bar{y})(\exists \bar{z})\big(x = y_0 \wedge x = z_0 \wedge y_n \not\approx z_m \wedge\\
&\qquad \mathrm{Tr}_{\bar{y}}(\alpha) \wedge \mathrm{Tr}_{\bar{z}}(\beta)\big)\\
\mathrm{Tr}_{\bar{y}}(\alpha) &= \bigwedge_{i=0}^{n-1} o_{i+1}(y_i, y_{i+1}) \wedge \bigwedge_{i=0}^{n} \mathrm{Tr}_{y_i}(\varphi_i),
\end{aligned}$$

where $\bar{y} = y_0, \ldots, y_n$ and $\bar{z} = z_0, \ldots, z_m$, and are fresh when quantified in the fourth and fifth definition;

$$\alpha = [\varphi_0]o_1[\varphi_1]o_2[\varphi_2]o_3 \cdots o_n[\varphi_n];$$
$$\beta = [\psi_0]o'_1[\psi_1]o'_2[\psi_2]o'_3 \cdots o'_m[\psi_m];$$

$o_i, o'_i \in \{\downarrow, \uparrow\}$; $o_j(u,v)$ represents $u \rightsquigarrow v$ if $o_j = \downarrow$, and $v \rightsquigarrow u$ otherwise.

Proposition 6.2. *For $\varphi \in XPath^{\updownarrow}_{=}$ we have $\mathcal{T}, u \models \varphi$ iff $\mathcal{T} \models \mathrm{Tr}_x(\varphi)(u)$.*

6.2 Downward XPath

Let $\mathrm{FO}(\sigma)$ be the set of first-order formulas over a given signature σ, and let $\mathcal{C}$ be a class of σ-models. An $\mathrm{FO}(\sigma)$-formula $\varphi(x)$ is **ℓ-local** if for all data trees $\mathcal{T}$ and $u \in T$, we have $\mathcal{T} \models \varphi(u) \Leftrightarrow \mathcal{T}|_\ell u \models \varphi(u)$. Finally, for $\varphi \in \mathrm{FO}(\sigma)$ let $\mathrm{qr}(\varphi)$ be its quantifier rank, i.e., the depth of nesting of its quantifiers.

Observe that the following result has two readings: one classical, and one restricted to finite models.

Theorem 6.3 (Characterization). *Let $\varphi(x) \in \mathrm{FO}(\sigma)$. The following are equivalent:*

(i) φ is $\leftrightarrows^{\downarrow}$-invariant over [finite] data-trees;

(ii) φ is logically equivalent over [finite] data-trees to an ℓ-$XPath^{\downarrow}_{=}$-formula, where $\ell = 2^{\mathrm{qr}(\varphi)} - 1$.

Proof. The implication (ii) $\Rightarrow$ (i) follows straightforwardly from Theorem 3.7. The proof of (i) $\Rightarrow$ (ii) goes as follows: First, we show that any $\leftrightarrows^{\downarrow}$-invariant $\varphi(x) \in \mathrm{FO}(\sigma)$ is ℓ-local for $\ell = 2^{\mathrm{qr}(\varphi)} - 1$ (Proposition 6.4). Then, we prove that any $\leftrightarrows^{\downarrow}$-invariant $\varphi(x) \in \mathrm{FO}(\sigma)$ that is ℓ-local is $\leftrightarrows^{\downarrow}_{\ell}$-invariant(Proposition B.2 in the Appendix). Finally, we show that any $\mathrm{FO}(\sigma)$-definable property which is $\leftrightarrows^{\downarrow}_{\ell}$-invariant is definable in ℓ-$\mathrm{XPath}^{\downarrow}_{=}$(Proposition B.3 in the Appendix). □

Proposition 6.4. *Any $\leftrightarrows^{\downarrow}$-invariant $\varphi(x) \in \mathrm{FO}(\sigma)$ over [finite] data-trees is ℓ-local for $\ell = 2^{\mathrm{qr}(\varphi)} - 1$.*

Proof. We follow Otto's proof [20]. Assume that $\varphi(x) \in \mathrm{FO}(\sigma)$ is $\leftrightarrows^{\downarrow}$-invariant, let $q = \mathrm{qr}(\varphi)$, and put $\ell = 2^q - 1$. Given a data tree $\mathcal{T}$ and $u \in T$ it suffices to show the existence of data trees $\mathcal{T}'$ and $\mathcal{T}''$, with corresponding elements $u' \in T'$ and $u'' \in T''$ such that

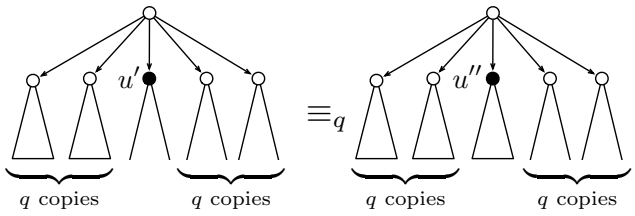

Figure 4: Definition of $\mathcal{T}', u'$ and $\mathcal{T}'', u''$.

(a) $\mathcal{T}', u' \leftrightarrows^{\downarrow} \mathcal{T}, u$,

(b) $\mathcal{T}'', u'' \leftrightarrows^{\downarrow} (\mathcal{T}|_\ell u), u$, and

(c) $\mathcal{T}', u' \equiv_q \mathcal{T}'', u''$.

Indeed, from the above conditions it follows that

$$\begin{aligned}
\mathcal{T} \models \varphi(u) &\text{ iff } \mathcal{T}' \models \varphi(u') && ((a) \text{ and } \leftrightarrows^{\downarrow}\text{-inv. of } \varphi)\\
&\text{iff } \mathcal{T}'' \models \varphi(u'') && (c)\\
&\text{iff } (\mathcal{T}|_\ell u) \models \varphi(u), && ((b) \text{ and } \leftrightarrows^{\downarrow}\text{-inv. of } \varphi)
\end{aligned}$$

and hence φ is ℓ-local. By Proposition 3.3 one may assume that $u \in T$ is the root of $\mathcal{T}$.

We define $\mathcal{T}'$ and $\mathcal{T}''$, as structures that are disjoint copies of sufficiently many isomorphic copies of $\mathcal{T}$ and $\mathcal{T}|_\ell u$, respectively, all tied together by some common root. Both structures have q isomorphic copies of both $\mathcal{T}$ and $\mathcal{T}|_\ell u$, and only distinguish themselves by the nature of the one extra subtree, in which u' and u'' live, respectively: u' is the root of one of the copies of $\mathcal{T}$ and u'' is the root of one of the copies of $\mathcal{T}|_\ell u$. We indicate the two structures in the diagram of Figure 4, with distinguished elements u' and u'' marked by $\bullet$; the open cones stand for copies of $\mathcal{T}$, the closed cones for copies of $\mathcal{T}|_\ell u$. The new isomorphic copies have the same data values as the original one. The new root has an arbitrary, fixed, data value and label.

By Proposition 3.4, it is straightforward that conditions (a) and (b) are satisfied. Condition (c) is true because one can exhibit a strategy for player **II** in the q-round Ehrenfeucht-Fraïssé game on structures $\mathcal{T}'$ and $\mathcal{T}''$. The strategy is exactly the same used in [20]. □

6.3 Vertical XPath

The analog of Theorem 6.3 fails for $\mathrm{XPath}^{\updownarrow}_{=}$:

Lemma 6.5. *The $\mathrm{FO}(\sigma)$-formula*

$$(\exists x)\ P_a(x)$$

is $\leftrightarrows^{\updownarrow}$-invariant though not logically equivalent over [finite] data-trees to any $XPath^{\updownarrow}_{=}$-formula.

Hence $\mathrm{XPath}^{\updownarrow}_{=}$ is not the fragment of $\mathrm{FO}(\sigma)$ which is $\leftrightarrows^{\updownarrow}$-invariant over [finite] data-trees. However, the following analog of Proposition B.3 (needed for the proof of Theorem 6.3) still holds for the case of $\mathrm{XPath}^{\updownarrow}_{=}$:

Proposition 6.6. *Let $k' = k \cdot (r+s+2)$. If $\varphi(x) \in \mathrm{FO}(\sigma)$ is $\leftrightarrows^{\updownarrow}_{r,s,k'}$-invariant over [finite] data-trees, then there is $\psi \in (r,s,k)$-$XPath^{\updownarrow}_{=}$ such that $\mathrm{Tr}_x(\psi)$ is logically equivalent to φ over [finite] data-trees.*

Notice that the counterexample in Lemma 6.5 is an unrestricted, existential formula. One may wonder if it might be possible to extend the expressive power of $\text{XPath}^{\updownarrow}_{=}$ to accout for unrestricted quantification. The natural candidate would be the modal operator E (usually known as the existential modality) which, intuitively, let us express that there is some node in the model where a formula holds. But even with the additional expressive power provided by E the analog of Theorem 6.3 fails. Formally, consider the logic $\text{XPath}^{\updownarrow E}_{=}$, which results from adding the operator E to $\text{XPath}^{\updownarrow}_{=}$ with the following semantics: $[\![E\varphi]\!]^{\mathcal{T}} = T$ if $[\![\varphi]\!]^{\mathcal{T}} \neq \emptyset$, and $[\![E\varphi]\!]^{\mathcal{T}} = \emptyset$ otherwise.

The following lemma shows a counterexample to the analog of Theorem 6.3, showing that $\text{XPath}^{\updownarrow E}_{=}$ is not the fragment of $\text{FO}(\sigma)$ $\leftrightarrows^{\updownarrow}$-invariant over [finite] data-trees.

LEMMA 6.7. *The* FO(σ)*-formula*

$$(\exists y, z)\ [y \approx z \wedge P_a(y) \wedge P_b(z)]$$

is $\leftrightarrows^{\updownarrow}$-invariant though not logically equivalent over [finite] data-trees to any $\text{XPath}^{\updownarrow E}_{=}$-formula.

7. APPLICATIONS

We devote this section to exemplify how the model theoretic tools we developed can be used to show expressiveness results for $\text{XPath}_{=}$. We do not intend to be comprehensive; rather we will exhibit a number of different results that show possible uses of the notions of bisimulation we introduced.

7.1 Expressiveness hierarchies

Define $\equiv^{\downarrow}_{\ell,k}$ as the equivalence $\equiv^{\downarrow}_{\ell}$ restricted to formulas of nesting depth at most k, that is, $\mathcal{T}, u \equiv^{\downarrow}_{\ell,k} \mathcal{T}', u'$ iff for all $\varphi \in \text{XPath}^{\downarrow}_{=}$ such that $\text{dd}(\varphi) \leq \ell$ and $\text{nd}(\varphi) \leq k$ we have $\mathcal{T}, u \models \varphi$ iff $\mathcal{T}', u' \models \varphi$. Define a more fine-grained notion of bisimulation in a similar way. We say that $u \in T$ and $u' \in T'$ are **(ℓ, k)-bisimilar for $\text{XPath}^{\downarrow}_{=}$** (notation: $\mathcal{T}, u \leftrightarrows^{\downarrow}_{\ell,k} \mathcal{T}', u'$) if there is a family of relations $(Z_{j,t})_{j\leq\ell, t\leq k}$ in $T \times T'$ such that $uZ_{\ell,k}u'$ and for all $j \leq \ell$, $t \leq k$, $x \in T$ and $x' \in T'$ we have

- **Harmony:** If $xZ_{j,t}x'$ then $\mathit{label}(x) = \mathit{label}(x')$.
- **Zig:** If $xZ_{j,t}x'$, $x \xrightarrow{n} v$ and $x \xrightarrow{m} w$ with $n, m \leq j$ then there are $v', w' \in T'$ such that $x' \xrightarrow{n} v'$, $x' \xrightarrow{m} w'$ and
  1. $\mathit{data}(v) = \mathit{data}(w) \Leftrightarrow \mathit{data}(v') = \mathit{data}(w')$,
  2. if $t > 0$, $(\xrightarrow{i} v)\, Z_{j-n+i,t-1}\, (\xrightarrow{i} v')$ for all $0 \leq i < n$, and
  3. if $t > 0$, $(\xrightarrow{i} w)\, Z_{j-m+i,t-1}\, (\xrightarrow{i} w')$ for all $0 \leq i < m$.
- **Zag:** If $xZ_{j,t}x'$, $x' \xrightarrow{n} v'$ and $x' \xrightarrow{m} w'$ with $n, m \leq j$ then there are $v, w \in T$ such that $x \xrightarrow{n} v$, $x \xrightarrow{m} w$ and items 1, 2 and 3 above are verified.

Following the same ideas used in Propositions 3.8 and 3.9, it is easy to show that (ℓ, k)-bisimulations characterize (ℓ, k)-equivalence.

PROPOSITION 7.1. *$\mathcal{T}, u \leftrightarrows^{\downarrow}_{\ell,k} \mathcal{T}', u'$ iff $\mathcal{T}, u \equiv^{\downarrow}_{\ell,k} \mathcal{T}', u'$.*

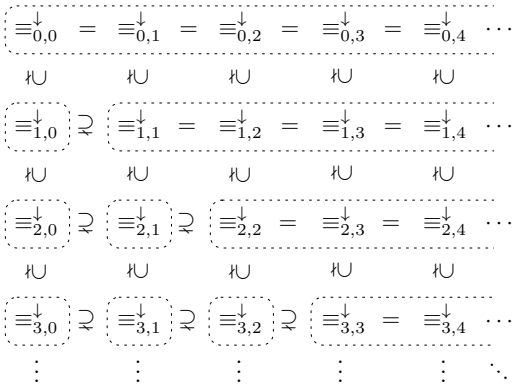

Figure 5: Hierarchy of $\text{XPath}^{\downarrow}_{=}$.

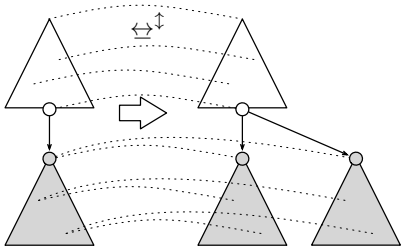

Figure 6: Closure under subtree replication.

The following theorem —proved in the Appendix using the bisimulation notion introduced above— characterizes when an increase in nesting depth results in an increase in expressive power (see Figure 5). We conjecture that a similar hierarchy holds in the absence of data values, but this is not a direct consequence of our result.

THEOREM 7.2. *For all $\ell, k \geq 0$, $i \geq 1$,*

$$\equiv^{\downarrow}_{\ell,0} \supsetneq \equiv^{\downarrow}_{\ell,1} \supsetneq \cdots \supsetneq \equiv^{\downarrow}_{\ell,\ell} = \equiv^{\downarrow}_{\ell,\ell+i}, \text{ and}$$
$$\equiv^{\downarrow}_{\ell,k} \supsetneq \equiv^{\downarrow}_{\ell+i,k}.$$

7.2 Safe operations on models

Bisimulations can also be used to show that certain operations on models preserve truth. Such operations are usually called *safe* for a given logic, as they can be applied to a model without changing the truth values of any formula in the language. Proposition 3.3, for example, is already an example of this kind of results showing that the class of models of a formula is closed under sub-model generation. We will now show a more elaborate example.

We say that $\mathcal{T}'$ is a **subtree replication** of $\mathcal{T}$, if $\mathcal{T}'$ is the result of inserting $\mathcal{T}|x$ into $\mathcal{T}$ as a sibling of x, where x is any node of $\mathcal{T}$ different from the root. Figure 6 gives a schematic representation of this operation.

PROPOSITION 7.3. *$\text{XPath}^{\updownarrow\downarrow^*}_{=}$ is closed under subtree replication, i.e. if $\mathcal{T}'$ is a subtree replication of $\mathcal{T}$, and $u \in T$ then $\mathcal{T}', u \equiv^{\updownarrow\downarrow^*} \mathcal{T}, u$.*

PROOF. Suppose that $x \in T$ is not the root of $\mathcal{T}$, and that $\mathcal{T}'$ is the result of inserting $\mathcal{T}|x$ into $\mathcal{T}$ as a sibling of x. Let us call $\mathcal{T}_x$ to the new copy of $\mathcal{T}|x$ inserted into $\mathcal{T}'$, and let X be the set of nodes of $\mathcal{T}|x$. Furthermore, if $v \in X$

then v_x is the corresponding node of $\mathcal{T}_x$. Nodes v and v_x have the same label and data value, and the position of v in $\mathcal{T}|x$ coincides with the position of v_x in $\mathcal{T}_x$.

By Theorem 5.1, it suffices to verify that $\mathcal{T}, u \leftrightarrows^{\updownarrow} \mathcal{T}', u$ via $Z \subseteq T \times T'$ defined by:

$$Z = \{(y, y) \mid y \in T\} \cup \{(v, v_x) \mid v \in X\}$$

(Z is depicted as dotted lines in Figure 6). □

7.3 Non-expressivity results

Finally, we will use bisimulation to show the expressivity limits of different fragments of XPath. Let $key(a)$ be the property stating that every node with label a has a different data value. Let $fk(a, b)$ (for *foreign key*) be the property $(\forall x)[P_a(x) \Rightarrow (\exists y)[P_b(y) \wedge x \sim y]]$.

PROPOSITION 7.4.

1. *$key(a)$ is not expressible in $XPath_{=}^{\updownarrow\updownarrow^*}$.*
2. *$fk(a, b)$ is expressible in $XPath_{=}^{\updownarrow\updownarrow^*}$ but it is not expressible in $XPath_{=}^{\downarrow\downarrow^*}$ or $XPath_{=}^{\updownarrow\updownarrow^{*+}}$.*

PROOF. The first item follows from Proposition 7.3. Since the logic is closed under subtree replication, the trees of Figure 7 are equivalent. As $key(a)$ holds in one and not in the other, the statement follows.

For the second item, it is easy to see that $fk(a, b)$ is expressible with the formula $\neg\langle\uparrow^*\downarrow_*[a \wedge \neg\langle\varepsilon = \uparrow^*\downarrow_*[b]\rangle]\rangle$. However, this property cannot be expressed in $\text{XPath}_{=}^{\downarrow\downarrow^*}$ because the models $\mathcal{T}$ and $\mathcal{T}'$ in Figure 8 are bisimilar for $\text{XPath}_{=}^{\downarrow}$ via Z, depicted as dotted lines. Since $\mathcal{T}, x$ satisfies $fk(a, b)$ but $\mathcal{T}', x'$ does not, from Theorem 5.1 it follows that $fk(a, b)$ is not expressible in $\text{XPath}_{=}^{\downarrow\downarrow^*}$.

Finally, suppose there exists $\psi \in \text{XPath}_{=}^{\updownarrow\updownarrow^{*+}}$ expressing $fk(a, b)$. Since $\mathcal{T}$ is a substructure of $\mathcal{T}'$ we have $\mathcal{T}, x \rightrightarrows^{\updownarrow} \mathcal{T}', x$ by Lemma 4.2. By Theorem 5.1(2) and the fact that $\mathcal{T}, x \models \psi$, we have $\mathcal{T}', x \models \psi$, which is a contradiction. □

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.

PROPOSITION 7.5.

1. *$dist_3$ is expressible in $XPath_{=}^{\updownarrow}$;*
2. *$dist_3$ is not expressible in $XPath_{=}^{\downarrow\downarrow^*}$;*
3. *neither $dist_3$ nor its complement can be expressed in $XPath_{=}^{\updownarrow\updownarrow^{*+}}$.*

PROOF. For 1, one can check that $\mathcal{T}, x \models \varphi$ iff $\mathcal{T}, x$ satisfies $dist_3$, for $\varphi = \langle\varepsilon \neq \downarrow\downarrow[\langle\varepsilon \neq \uparrow\rangle]\rangle$.

Let us see 2. Consider the data trees $\mathcal{T}, x$ and $\mathcal{T}', x'$ depicted in Figure 9. It is straightforward that $\mathcal{T}, x$ satisfies $dist_3$ and $\mathcal{T}', x'$ does not.

Let v_1' and v_2' be the leaves of $\mathcal{T}'$ and let v be the only node of $\mathcal{T}$ with data value 3. One can check that $\mathcal{T}, x \leftrightarrows^{\downarrow} \mathcal{T}', x'$ via $Z \subseteq T \times T'$ defined by

$$\begin{aligned} Z \;=\; & \{\langle u, u'\rangle \mid h(u) = h(u') \wedge data(u) = data(u')\} \cup \\ & \{\langle v, v_1'\rangle, \langle v, v_2'\rangle\}, \end{aligned}$$

where $h(y)$ denotes the height of y, i.e., the distance from y to the root of the corresponding tree (Z is depicted as dotted lines in Figure 9). Since $\mathcal{T}, x$ satisfies $dist_3$ but $\mathcal{T}', x'$ does not, from Theorem 5.1 it follows that $dist_3$ is not expressible in $\text{XPath}_{=}^{\downarrow\downarrow^*}$.

For 3, one can verify that $\mathcal{T}, x \rightrightarrows^{\updownarrow} \mathcal{T}', x'$ via Z as defined above. If $dist_3$ were definable in $\text{XPath}_{=}^{\updownarrow\updownarrow^{*+}}$ via ψ and the fact that $\mathcal{T}, x \models \psi$, by Theorem 5.1(2) we would have $\mathcal{T}', x' \models \psi$, and this is a contradiction.

Let $\overline{dist_3}$ denote the complement of $dist_3$, i.e., $\overline{dist_3}(x)$ iff for all y, z so that $x{\rightarrow}y{\rightarrow}z$, we have that x, y, z do not have pairwise distinct data values. Now $\mathcal{T}', x'$ satisfies $\overline{dist_3}$ and $\mathcal{T}, x$ does not. Since $\mathcal{T}'$ is a substructure of $\mathcal{T}$, by an argument analog to the one used in the proof of Proposition 7.4-2, we conclude that $\overline{dist_3}$ is not expressible in $\text{XPath}_{=}^{\updownarrow\updownarrow^{*+}}$. □

8. DISCUSSION

In this article we studied model theoretic properties of XPath over both finite and arbitrary data trees using bisimulations. One of the main results we discuss is the characterization of the downward and vertical fragments of XPath as the fragments of first-order logic which are invariant under suitable notions of bisimulation. This can be seen as a first step in the larger program of studying the model theory and expressiveness of XPath with data values and, more generally, of logics on data trees. It would be interesting to study notions of bisimulation with only descendant; or characterizations of XPath with child and descendant, as a fragment of FO with the descendant relation on data trees. We did not considered XPath with horizontal navigation between siblings, such as the axes `next-sibling` and `previous-sibling`. In fact, adding these axes results in a fragment that is somewhat less interesting since the adequate bisimulation notion on finite data trees corresponds precisely to data tree isomorphism modulo renaming of data values.

In Section 7 we show a number of concrete application of the model theoretic tools we developed, discussing both expressivity and non-expressivity results. We also show examples of operations which are safe for a given XPath fragment. It would be worthwhile to devise other model operations that preserve truth of XPath formulas as we show is the case for *subtree replication*.

An important application of bisimulation is as a minimization method: given a data tree $\mathcal{T}_1$ we want to find a data tree $\mathcal{T}_2$, as small as possible, so that $\mathcal{T}_1$ and $\mathcal{T}_2$ are bisimilar for some fragment $\mathcal{L}$ of XPath. Since $\mathcal{L}$ cannot distinguish between $\mathcal{T}_1$ and $\mathcal{T}_2$, we can use $\mathcal{T}_2$ as representative of $\mathcal{T}_1$ while the expressive power of $\mathcal{L}$ is all that is required by a given application. The complexity of several inference tasks (e.g., model checking) depends directly on the model size. This is why in some cases it may be profitable to first apply a minimization step. The existence of efficient minimization algorithms is intimately related to bisimulations: we can minimize a data tree $\mathcal{T}$ by partitioning it in terms of its coarsest auto-bisimulation. We plan to design and implement algorithms for data tree minimization using bisimulation and investigate their computational complexity.

References

[1] M. Benedikt, W. Fan, and F. Geerts. XPath satisfiability in the presence of DTDs. *J. of the ACM*, 55(2):1–79, 2008.

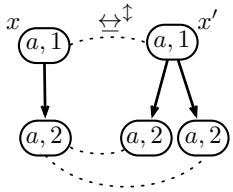

Figure 7: *key*(a) not in XPath$_=^{\updownarrow\updownarrow^*}$.

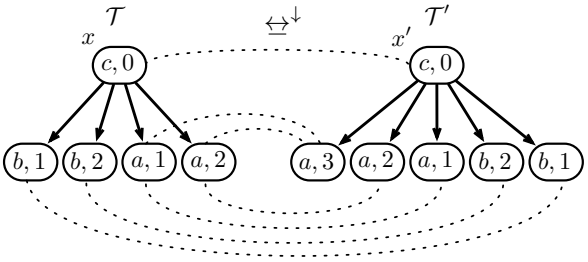

Figure 8: *fk*(a, b) not in XPath$_=^{\downarrow\downarrow_*}$.

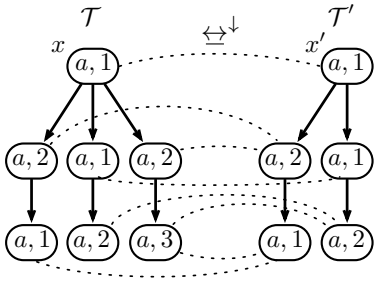

Figure 9: *dist*$_3$ not in XPath$_=^{\downarrow\downarrow_*}$.

[2] M. Benedikt and C. Koch. XPath leashed. *ACM Comput. Sur.*, 41(1), 2008.

[3] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*, volume 53 of *Cambridge Tracts Theoret. Comput. Sci.* Cambridge University Press, 2001.

[4] M. Bojańczyk, A. Muscholl, T. Schwentick, and L. Segoufin. Two-variable logic on data trees and XML reasoning. *J. of the ACM*, 56(3):1–48, 2009.

[5] M. Bojańczyk and P. Parys. XPath evaluation in linear time. *J. of the ACM*, 58(4):17, 2011.

[6] J. Clark and S. DeRose. XML path language (XPath). Website, 1999. W3C Recommendation. `http://www.w3.org/TR/xpath`.

[7] A. Dawar and M. Otto. Modal characterisation theorems over special classes of frames. *Ann. Pure Appl. Logic*, 161(1):1–42, 2009.

[8] D. Figueira. *Reasoning on Words and Trees with Data*. PhD thesis, Laboratoire Spécification et Vérification, ENS Cachan, France, 2010.

[9] D. Figueira. Decidability of downward XPath. *ACM Trans. Comput. Log.*, 13(4), 2012.

[10] D. Figueira and L. Segoufin. Bottom-up automata on data trees and vertical XPath. In *Int. Symp. on Theoretical Aspects of Computer Science (STACS'11)*, volume 9 of *LIPIcs*, pages 93–104. Leibniz-Zentrum für Informatik, 2011.

[11] M. Forti and F. Honsell. Set theory with free construction principles. *Annali Scuola Normale Superiore, Pisa*, X(3):493–522, 1983.

[12] V. Goranko and M. Otto. Model theory of modal logic. In J. Van Benthem P. Blackburn and F. Wolter, editors, *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*, chapter 5, pages 249–329. Elsevier, 2007.

[13] G. Gottlob, C. Koch, and R. Pichler. Efficient algorithms for processing XPath queries. *ACM Trans. Database Systems*, 30(2):444–491, 2005.

[14] Marc Gyssens, Jan Paredaens, Dirk Van Gucht, and George H. L. Fletcher. Structural characterizations of the semantics of xpath as navigation tool on a document. In *PODS*, pages 318–327. ACM, 2006.

[15] D. Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic. Vol. II*, volume 165 of *Synthese Library*, pages 497–604. D. Reidel Publishing Co., Dordrecht, 1984. Extensions of classical logic.

[16] N. Kurtonina and M. de Rijke. Simulating without negation. *J. Logic Comput.*, 7:503–524, 1997.

[17] M. Marx. XPath with conditional axis relations. In *Int. Conf. on Extending Database Technology (EDBT'04)*, volume 2992 of *LNCS*, pages 477–494. Springer, 2004.

[18] M. Marx and M. de Rijke. Semantic characterizations of navigational XPath. *SIGMOD Record*, 34(2):41–46, 2005.

[19] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *LNCS*. Springer, 1980.

[20] M. Otto. Elementary proof of the Van Benthem-Rosen characterisation theorem. Technical Report 2342, Fachbereich Mathematik, Technische Universität Darmstadt, 2004.

[21] M. Otto. Modal and guarded characterisation theorems over finite transition systems. *Ann. Pure Appl. Logic*, 130(1-3):173–205, 2004.

[22] M. Otto. Bisimulation invariance and finite models. In *Logic Colloquium'02*, volume 27 of *Lect. Notes Log.*, pages 276–298, 2006.

[23] D. Park. Concurrency and automata on infinite sequences. In *Theoret. Comput. Sci.*, volume 104 of *LNCS*, pages 167–183. Springer, 1981.

[24] E. Rosen. Modal logic over finite structures. *J. Logic Lang. Inform.*, 6(4):427–439, 1997.

[25] Davide Sangiorgi. On the origins of bisimulation and coinduction. *ACM Transactions on Programming Languages and Systems*, 31(4), 2009.

[26] J. van Benthem. *Modal Correspondence Theory*. PhD thesis, Universiteit van Amsterdam, 1976.

APPENDIX

A. PROOFS OF SECTION 3

Given a path expression α, the **navigation of** α (notation: nav(α)) is the string of $\{\uparrow, \downarrow\}^*$ that results from removing all node expressions $[\psi]$ and ε from α. For example, nav$(\downarrow[\langle\uparrow\rangle]\downarrow[\downarrow = \uparrow\rangle]\uparrow[b]) = \downarrow\downarrow\uparrow$.

LEMMA A.1. *Let α be a path expression of XPath$_=^{\downarrow\downarrow_*}$. Let $x \xrightarrow{n} v$ and $x' \xrightarrow{n} v'$ such that $(x, v) \in [\![\alpha]\!]^{\mathcal{T}}$ and $(x', v') \notin [\![\alpha]\!]^{\mathcal{T}'}$.*

Then there is a subformula φ of α and $k \in \{0,\dots,n\}$ such that $\mathcal{T}, (\xrightarrow{k} v) \models \varphi$ and $\mathcal{T}', (\xrightarrow{k} v') \not\models \varphi$.

PROOF. Let $x = v_0 \to v_1 \to \cdots \to v_n = v$ and $x' = v'_0 \to v'_1 \to \cdots \to v'_n = v'$. We proceed by induction on $|\alpha|$. If $\alpha = \varepsilon$ then $x = v$ and so $n = 0$. Hence $x' = v'$ and $(x', v') \in [\![\alpha]\!]^{\mathcal{T}'}$, which contradicts the hypothesis, and thus the statement is trivially true. If $\alpha = {\downarrow}$ then $x \to v$ and so $n = 1$. Hence $x' \to v'$ and $(x', v') \in [\![\alpha]\!]^{\mathcal{T}'}$. This case is also trivial. The case $\alpha = {\downarrow_*}$ is similar.

Suppose $\alpha = [\psi]$. Since $(x', v') \notin [\![\alpha]\!]^{\mathcal{T}'}$, we have $x' = v'$ and $\mathcal{T}', v' \not\models \psi$. Taking $k = 0$ and $\varphi = \psi$ the statement holds. Observe that ψ is a subformula of α.

Suppose $\alpha = \beta\gamma$. Then there is k such that $(x, v_k) \in [\![\beta]\!]^{\mathcal{T}}$ and $(v_k, v) \in [\![\gamma]\!]^{\mathcal{T}}$. Since $(x', v') \notin [\![\alpha]\!]^{\mathcal{T}'}$, we have $(x', v'_k) \notin [\![\beta]\!]^{\mathcal{T}'}$ or $(v'_k, v') \notin [\![\gamma]\!]^{\mathcal{T}'}$. In either case, apply inductive hypothesis straightforwardly. □

PROPOSITION 3.1. *$\equiv^{\downarrow}_{\ell}$ has finite index.*

PROOF. We show by induction on ℓ that there are finitely many non-equivalent formulas of downward depth at most ℓ, and finitely many non-equivalent path expressions of downward depth at most ℓ.

For the base case, any formula of downward depth 0 is a Boolean combination of labels, and hence there are finitely many non-equivalent of them. Any path expression of downward depth 0 is equivalent to $[\varphi]$ for $\mathrm{dd}(\varphi) = 0$, and hence there are finitely many non-equivalent of them.

For the induction, any formula of downward depth $\ell + 1$ is a boolean combination of labels or formulas of the form $\langle\alpha\rangle$, $\langle\alpha = \beta\rangle$ or $\langle\alpha \neq \beta\rangle$, where $\mathrm{dd}(\alpha), \mathrm{dd}(\beta) \leq \ell + 1$, so it suffices to show that there are finitely many non-equivalent path expressions of downward depth at most $\ell + 1$. Let α be such that $\mathrm{dd}(\alpha) \leq \ell + 1$. By Proposition 6.1, α is either equivalent to a path expression of the form $[\psi]$ or of the form $[\psi] \downarrow \beta$, where $\mathrm{dd}(\psi), \mathrm{dd}(\beta) \leq \ell$. By inductive hypothesis there are finitely many non-equivalent ψ's and β_i's, and hence finitely many non-equivalent α's. □

COROLLARY 3.2. *$\{\mathcal{T}', u' \mid \mathcal{T}, u \equiv^{\downarrow}_{\ell} \mathcal{T}', u'\}$ is definable by an ℓ-XPath$^{\downarrow}_{=}$-formula $\chi_{\ell,\mathcal{T},u}$.*

PROOF. Consider the conjunction of all ℓ-XPath$^{\downarrow}_{=}$ formulas φ such that $\mathcal{T}, u \models \varphi$. By Proposition 3.1, up to logical equivalence, there are finitely many such φ's, and hence the conjunction is equivalent to a finite one. Define $\chi_{\ell,\mathcal{T},u}$ as this finite conjunction. □

THEOREM 3.7.

1. *$\mathcal{T}, u \leftrightarrows^{\downarrow} \mathcal{T}', u'$ implies $\mathcal{T}, u \equiv^{\downarrow} \mathcal{T}', u'$. The converse also holds when $\mathcal{T}$ and $\mathcal{T}'$ are finitely branching.*
2. *$\mathcal{T}, u \leftrightarrows^{\downarrow}_{\ell} \mathcal{T}', u'$ iff $\mathcal{T}, u \equiv^{\downarrow}_{\ell} \mathcal{T}', u'$.*

PROOF. Item 2 is a direct consequence of Propositions 3.8 and 3.9. The argument for item 1 is similar to the one of the aforementioned propositions, but working with a single Z instead of $(Z_i)_{i \leq \ell}$. For the converse implication, define Z by xZx' iff $\mathcal{T}, x \equiv^{\downarrow} \mathcal{T}', x'$. The conjunctions in (1) are then finite because $\mathcal{T}'$ is finitely branching, and so P is finite (the fact that $\mathcal{T}$ is finitely branching is used to show that Zag is satisfied). □

PROPOSITION 3.10. *Let $\varphi \in (r, s, k)$-XPath$^{\updownarrow}_{=}$. There is $\varphi^{\uparrow\downarrow} \in$ XPath$^{\updownarrow}_{=}$ in up-down normal form such that*

1. *$\varphi^{\uparrow\downarrow} \not\equiv \varphi$;*
2. *$\mathrm{vd}(\varphi^{\uparrow\downarrow}) = (r, s)$; and*
3. *$\mathrm{nd}(\varphi^{\uparrow\downarrow}) \leq k \cdot (r + s + 2)$.*

PROOF. The idea is that we can factorize any path in the tree going down and up as a node tests in the expression. Consider for instance the expression $\alpha = {\uparrow}{\downarrow}[a]{\uparrow}{\uparrow}{\downarrow}$. It is immediate that α is equivalent to the up-down expression $[\langle{\uparrow}[\langle{\downarrow}[a]\rangle]\rangle]{\uparrow}{\uparrow}{\downarrow}$, which is in up-down normal form.

We use the following directed equivalences to translate any path expression into an equivalent up-down expression.

$$\varepsilon\gamma \equiv^{\updownarrow} \gamma \qquad (\varepsilon)$$

$$\alpha[\psi_1][\psi_2]\beta \equiv^{\updownarrow} \alpha[\psi_1 \wedge \psi_2]\beta \qquad (\textit{merge})$$

$$\alpha\ \xi_{-n}{\downarrow}\cdots{\downarrow}\xi_{-1}{\downarrow}\xi_0{\uparrow}\xi_1{\uparrow}\cdots{\uparrow}\xi_n\ \beta \equiv^{\updownarrow} \alpha[\langle\xi_{-n}\xi_n{\downarrow}\cdots{\downarrow}\xi_{-1}\xi_1{\downarrow}\xi_0\rangle]\beta \qquad (\textit{factor})$$

$$\alpha\ \xi_n{\downarrow}\xi_{n-1}{\downarrow}\cdots{\downarrow}\xi_0 \equiv^{\updownarrow} \alpha\downarrow^n[\langle\xi_0{\uparrow}\xi_1{\uparrow}\ldots{\uparrow}\xi_n\rangle] \qquad (\textit{shift-r})$$

$$\xi_0{\uparrow}\xi_1{\uparrow}\cdots{\uparrow}\xi_n\ \beta \equiv^{\updownarrow} [\langle\xi_0{\uparrow}\xi_1{\uparrow}\cdots{\uparrow}\xi_n\rangle]{\uparrow}^n\ \beta \qquad (\textit{shift-l})$$

In the expressions above, each ξ_i is the empty string, or of the form ε or $[\varphi_1][\varphi_2]\ldots[\varphi_n]$, α and β can be any path expression, or the empty string, and γ is any path expression. The idea is that (*factor*) converts an expression that goes down n times and then up n times into a node expression, and when doing this, any test done in the i-th node when going down is merged with the $(n-i)$-th test when going up. For example, ${\downarrow}[\neg a]{\downarrow}[c]{\uparrow}[\neg b]{\uparrow} \equiv^{\updownarrow} [\langle{\downarrow}[\neg a][\neg b]{\downarrow}[c]\rangle]$. On the other hand, (*shift-r*) and (*shift-l*) group all the node tests in the lowest node in the expression, making use of the fact that the parent relation is functional. Thus, for example $[a]{\downarrow}[b]{\downarrow} \equiv^{\updownarrow} {\downarrow}{\downarrow}[\langle{\uparrow}[b]{\uparrow}[a]\rangle]$ and ${\uparrow}[a]{\uparrow}[b] \equiv^{\updownarrow} [\langle{\uparrow}[a]{\uparrow}[b]\rangle]{\uparrow}{\uparrow}$. It is thus clear that the left and right expressions above are semantically equivalent.

LEMMA A.2. *Let α be an XPath$^{\updownarrow}_{=}$-path expression with $\mathrm{vd}(\alpha) = (r, s)$ and $\mathrm{nd}(\alpha) = k$, Then there is an up-down path expression $\alpha^{\uparrow\downarrow}$ such that:*

1. *$\alpha^{\uparrow\downarrow} \equiv^{\updownarrow} \alpha$*
2. *$\mathrm{vd}(\alpha^{\uparrow\downarrow}) = (r, s)$, and*
3. *$\mathrm{nd}(\alpha^{\uparrow\downarrow}) \leq k + r + s + 1$.*

PROOF. We first apply rule (*factor*) as many times as possible. It is clear that if $\mathrm{nav}(\alpha)$ is of the form ${\uparrow}^n{\downarrow}^m$ for some $n, m \geq 0$ then rule (*factor*) cannot be applied and we are done. Hence, suppose $\mathrm{nav}(\alpha)$ contains the pattern ${\downarrow}{\uparrow}$. Let

$$\begin{aligned} \alpha &= \gamma_{\uparrow}\alpha_1\gamma_{\downarrow} \\ \alpha_1 &= \gamma_1 \underbrace{\xi^1_{-n_1}{\downarrow}\ldots{\downarrow}\xi^1_0{\uparrow}\ldots{\uparrow}\xi^1_{n_1}}_{\text{matches } (\textit{factor})} \\ &\qquad \gamma_2 \underbrace{\xi^2_{-n_2}{\downarrow}\ldots{\downarrow}\xi^2_0{\uparrow}\ldots{\uparrow}\xi^2_{n_2}}_{\text{matches } (\textit{factor})} \\ &\qquad\qquad \vdots \end{aligned}$$

$$\gamma_{m-1}\underbrace{\xi^m_{-n_m}\downarrow\ldots\downarrow\xi^m_0\uparrow\ldots\uparrow\xi^m_{n_m}}_{\text{matches }(factor)}\gamma_m,$$

where $\mathrm{nav}(\gamma_\uparrow), \mathrm{nav}(\gamma_m) \in \uparrow^*$, $\mathrm{nav}(\gamma_\downarrow), \mathrm{nav}(\gamma_1) \in \downarrow^*$, and ξ^i_j are the empty string, ε or $[\varphi^{i,j}_1][\varphi^{i,j}_2]\ldots[\varphi^{i,j}_{h_{i,j}}]$. Furthermore, assume that m is maximal (i.e., it is impossible to apply (*factor*) in any of the γ_i's) and that the length of each γ_i is minimal (i.e., it is not the case that $\mathrm{nav}(\gamma_i)$ ends with $\downarrow$ and that $\mathrm{nav}(\gamma_{i+1})$ begins with $\uparrow$). Observe that $\mathrm{nav}(\gamma_i) \in \uparrow^*\downarrow^*$. We apply rule (*factor*) in the $m-1$ marked places and obtain

$$\begin{aligned}\alpha_2 \;=\; & \gamma_1\underbrace{[\langle\xi^1_{-n_1}\xi^1_{n_1}\downarrow\cdots\downarrow\xi^1_{-1}\xi^1_1\downarrow\xi^1_0\rangle]}_{(factor)\text{ applied}}\\ & \gamma_2\underbrace{[\langle\xi^2_{-n_2}\xi^2_{n_2}\downarrow\cdots\downarrow\xi^2_{-1}\xi^2_1\downarrow\xi^2_0\rangle]}_{(factor)\text{ applied}}\\ & \vdots\\ & \gamma_{m-1}\underbrace{[\langle\xi^m_{-n_m}\xi^m_{n_m}\downarrow\cdots\downarrow\xi^m_{-1}\xi^m_1\downarrow\xi^m_0\rangle]}_{(factor)\text{ applied}}\gamma_m,\end{aligned}$$

Let $\mathrm{vd}(\mathrm{nav}(\alpha_1)) = (r_1, s_1)$. Since $\mathrm{nav}(\alpha) = \mathrm{nav}(\gamma_\uparrow\alpha_1\gamma_\downarrow)$ contains the pattern $\downarrow\uparrow$, we have that $r_1 > 0$. It can be shown that $\mathrm{vd}(\gamma_\uparrow\alpha_2\gamma_\uparrow) = (r, s)$, $\mathrm{nd}(\alpha_2) \le \mathrm{nd}(\alpha_1) + 1$, and $\mathrm{vd}(\mathrm{nav}(\alpha_2)) \le (r_1 - 1, s_1)$. If we repeat this procedure with α_2 and so on until we can no longer apply rule (*factor*), we end up with an up-down path expression α_f so that

1. $\alpha_f \equiv^\updownarrow \alpha_1$,
2. $\mathrm{vd}(\gamma_\uparrow\alpha_f\gamma_\downarrow) = (r, s)$, and
3. $\mathrm{nd}(\alpha_f) \le \mathrm{nd}(\alpha_1) + r_1$.

We can now apply (*shift-r*), (*shift-l*), (ε) and (*merge*) to $\gamma_\uparrow\alpha_f\gamma_\downarrow$ in order to obtain an up-down path expression $\alpha^{\uparrow\downarrow}$ satisfying the desired conditions:

1. $\alpha^{\uparrow\downarrow} \equiv^\updownarrow \alpha$
2. $\mathrm{vd}(\alpha^{\uparrow\downarrow}) = (r, s)$, and
3. $\mathrm{nd}(\alpha^{\uparrow\downarrow}) \le k + r_1 + 1 \le r + s + 1$.

This concludes the proof of Lemma A.2. □

Lemma A.3. *Let $\alpha^{\uparrow\downarrow}, \beta^{\uparrow\downarrow}$ be up-down path expressions and let $\varphi = \langle\alpha^{\uparrow\downarrow} \odot \beta^{\uparrow\downarrow}\rangle$ (for $\odot \in \{=, \neq\}$) with $\mathrm{vd}(\varphi) = (r, s)$ and $\mathrm{nd}(\varphi) = k$. Then there is an up-down path expression $\gamma^{\uparrow\downarrow}$ such that:*

1. *$\langle\varepsilon \odot \gamma^{\uparrow\downarrow}\rangle \equiv^\updownarrow \varphi$,*
2. *$\mathrm{vd}(\gamma^{\uparrow\downarrow}) = (r, s)$, and*
3. *$\mathrm{nd}(\gamma^{\uparrow\downarrow}) \le k + 1$.*

Proof. Let us analyse the case where

$$\begin{aligned}\alpha^{\uparrow\downarrow} &= [\psi_\alpha]\uparrow^{n_\alpha}\downarrow^{m_\alpha}[\tau_\alpha]\\ \beta^{\uparrow\downarrow} &= [\psi_\beta]\uparrow^{n_\beta}\downarrow^{m_\beta}[\tau_\beta]\end{aligned}$$

(the remaining cases being only simpler), where ψ_α, ψ_β, τ_α, τ_β are in up-down normal form. Suppose, without loss of generality, that $n_\alpha \le n_\beta$. Hence, we have $\langle\alpha^{\uparrow\downarrow} \odot \beta^{\uparrow\downarrow}\rangle \equiv^\updownarrow$

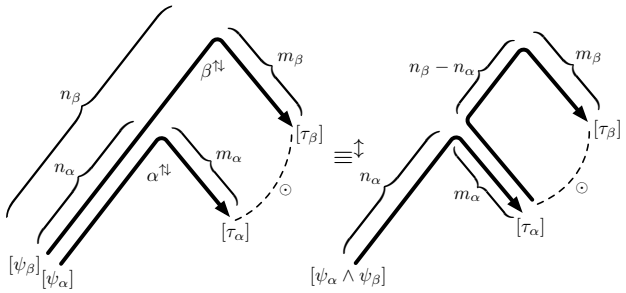

Figure 10: Normal form for data tests.

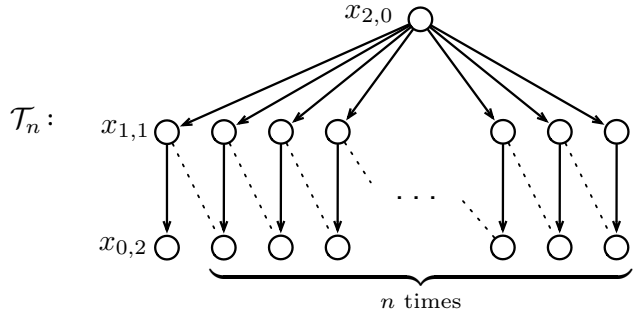

Figure 11: Model verifying ψ^j_i for all $i \ge n$ and not verifying φ_l for no $l < n$. Dotted lines represent equal data values.

$\langle\varepsilon \odot \gamma^{\uparrow\downarrow}\rangle$, where

$$\gamma^{\uparrow\downarrow} = [\psi_\alpha \wedge \psi_\beta]\uparrow^{n_\alpha}\downarrow^{m_\alpha}\ [\tau_\alpha \wedge \langle\varepsilon \odot \uparrow^{m_\alpha}\uparrow^{n_\beta - n_\alpha}\downarrow^{m_\beta}[\tau_\beta]\rangle].$$

It is clear that the formulas are equivalent (*cf.* Figure 10). Moreover, the right-hand formula has at most one more nesting than the left-hand formula, and its vertical depth is at most (r, s). This concludes the proof of Lemma A.3. □

By induction on φ, and using lemmas A.2 and A.3, one can show that there is $\varphi^{\uparrow\downarrow}$ as desired.

Proposition 3.11. *If $r + s \ge 2$ then $\equiv^\updownarrow_{r,s}$ has infinite index.*

Proof. We show that for every r, s so that $r + s = 2$ there is an infinite set of non-equivalent formulas $\{\psi^i_{r,s}\}_{i\ge0}$ of vertical depth (r, s). It thus follows that for every r, s so that $r + s \ge 2$, $\equiv^\updownarrow_{r,s}$ has infinite index.

Consider the following formulas.

$$\begin{aligned}\psi^0_{1,1} &= \langle\varepsilon = \uparrow\downarrow\downarrow\rangle & \psi^{i+1}_{1,1} &= \langle\varepsilon = \uparrow\downarrow[\psi^i_{1,1}]\downarrow\rangle\\ \psi^0_{0,2} &= \langle\uparrow = \uparrow\uparrow\downarrow\downarrow\rangle & \psi^{i+1}_{0,2} &= \langle\uparrow = \uparrow\uparrow\downarrow[\psi^i_{1,1}]\downarrow\rangle\\ \psi^0_{2,0} &= \langle\downarrow = \downarrow\downarrow\rangle & \psi^{i+1}_{2,0} &= \langle\downarrow = \downarrow[\psi^i_{1,1}]\downarrow\rangle\end{aligned}$$

Note that $\mathrm{vd}(\psi^n_{r,s}) = (r, s)$ and $\mathrm{nd}(\psi^n_{r,s}) = n$ for every n. The formula $\psi^n_{r,s}$ intuitively says that there is a chain of length n as depicted in Figure 11.

In the data tree $\mathcal{T}_n$ of the figure, we have that $\mathcal{T}_n, x_{r,s} \models \psi^n_{r,s}$ but $\mathcal{T}_n, x_{r,s} \not\models \psi^{n'}_{r,s}$ for any $n' > n$. Therefore, $\{\psi^i_{r,s}\}_{i\ge0}$ is an infinite set of non-equivalent formulas of vertical depth (r, s). □

PROPOSITION 3.12. $\equiv^{\updownarrow}_{r,s,k}$ *has finite index.*

PROOF. For any φ with $\mathrm{nd}(\varphi) = k$ and $\mathrm{vd}(\varphi) = (r,s)$, let $F(\varphi) = (k, r+s)$. Define F in a similar way for path expressions α. In this proof "finite" means finite up to logical equivalence. By Proposition 3.10 we can consider only formulas in up-down normal form.

We show that there are finitely many formulas φ in up-down normal form such that $F(\varphi) \leq (k,t)$, and that there are finitely many path expressions α in up-down normal form such that $F(\alpha) \leq (k,t)$. We use induction on the lexicographic ordering of (k,t). Observe that if $F(\varphi) = (k,t)$ then φ is a boolean combination of labels and formulas of the form $\langle \varepsilon = \alpha \rangle$, $\langle \varepsilon \neq \alpha \rangle$ or $\langle \alpha \rangle$, where $F(\alpha) \leq (k,t)$. Hence it suffices to show the statement for path expressions. If $F(\alpha) = (0,t)$ then α is either ε or $\uparrow^n\downarrow^m$, where $n, m \leq 2t$, so there are finitely many of them. If $F(\alpha) = (k+1, t)$, then α is $[\varphi_1]\uparrow^n\downarrow^m[\varphi_2]$, where $n, m \leq 2t$ and $\mathrm{nd}(\varphi_i) \leq k$ for $i = 1, 2$. Since $F(\varphi_i) <_{\mathrm{lex}} (k+1, t)$, by inductive hypothesis there are finitely many such φ_i's, and therefore α's. $\square$

COROLLARY 3.13. $\{\mathcal{T}', u' \mid \mathcal{T}, u \equiv^{\updownarrow}_{r,s,k} \mathcal{T}', u'\}$ *is definable by an* (r,s,k)*-XPath*$^{\updownarrow}_{=}$*-formula.*

PROOF. Similar to the proof of Corollary 3.2. $\square$

THEOREM 3.16.

1. $\mathcal{T}, u \leftrightarrows^{\updownarrow} \mathcal{T}', u'$ *implies* $\mathcal{T}, u \equiv^{\updownarrow} \mathcal{T}', u'$*. The converse also holds when* $\mathcal{T}$ *and* $\mathcal{T}'$ *are finitely branching.*
2. $\mathcal{T}, u \leftrightarrows^{\updownarrow}_{r,s,k\cdot(r+s+2)} \mathcal{T}', u'$ *implies* $\mathcal{T}, u \equiv^{\updownarrow}_{r,s,k} \mathcal{T}', u'$*.*
3. $\mathcal{T}, u \equiv^{\updownarrow}_{r,s,k} \mathcal{T}', u'$ *implies* $\mathcal{T}, u \leftrightarrows^{\updownarrow}_{r,s,k} \mathcal{T}', u'$*.*

PROOF. Items 2 and 3 are shown in Propositions A.4 and A.5.

The argument for item 1 is similar to the one of the aforementioned propositions, but working with a single Z instead of $(Z^{\hat{k}}_{\hat{r},\hat{s}})_{\hat{r},\hat{s},\hat{k}}$. For the converse implication, define Z by xZx' iff $\mathcal{T}, x \equiv^{\updownarrow} \mathcal{T}', x'$. The conjunctions in (2) are then finite because $\mathcal{T}'$ is finitely branching, and so P is finite (the fact that $\mathcal{T}$ is finitely branching is used for showing that Zag is satisfied). $\square$

PROPOSITION A.4. $\mathcal{T}, u \leftrightarrows^{\updownarrow}_{r,s,k\cdot(r+s+2)} \mathcal{T}', u'$ *implies* $\mathcal{T}, u \equiv^{\updownarrow}_{r,s,k} \mathcal{T}', u'$.

PROOF. We show that if $\mathcal{T}, u \leftrightarrows^{\updownarrow}_{r,s,k} \mathcal{T}', u'$ via

$$(Z^{\hat{k}}_{\hat{r},\hat{s}})_{\hat{r}+\hat{s}\leq r+s, \hat{k}\leq k}$$

then for all $n \leq \hat{s}$ and $m \leq \hat{r}+n$, for all φ in up-down normal form with $\mathrm{vd}(\varphi) \leq (\hat{r}, \hat{s})$, $\mathrm{nd}(\varphi) \leq \hat{k}$, for all upward expression $\alpha^{\uparrow}$ in up-down normal form, and for all downward expression $\alpha^{\downarrow}$ in up-down normal form with $\mathrm{vd}(\alpha^{\uparrow}), \mathrm{vd}(\alpha^{\downarrow}) \leq (\hat{r}, \hat{s})$, $\mathrm{nd}(\alpha^{\uparrow}), \mathrm{nd}(\alpha^{\downarrow}) \leq \hat{k}$:

1. If $xZ^{\hat{k}}_{\hat{r},\hat{s}}x'$ then $\mathcal{T}, x \models \varphi$ iff $\mathcal{T}', x' \models \varphi$.
2. If $y \xrightarrow{n} x$, $y' \xrightarrow{n} x'$, $x\, Z^{\hat{k}-1}_{\hat{r},\hat{s}}\, x'$, then $(x,y) \in [\![\alpha^{\uparrow}]\!]^{\mathcal{T}}$ iff $(x',y') \in [\![\alpha^{\uparrow}]\!]^{\mathcal{T}'}$.
3. If $y \xrightarrow{m} z$, $y' \xrightarrow{m} z'$, $z\, Z^{\hat{k}-1}_{\hat{r}',\hat{s}'}\, z'$ for $\hat{r}' = \hat{r} + n - m$, $\hat{s}' = \hat{s} - n + m$, then $(y,z) \in [\![\alpha^{\downarrow}]\!]^{\mathcal{T}}$ iff $(y',z') \in [\![\alpha^{\downarrow}]\!]^{\mathcal{T}'}$.

Hence, by Proposition 3.10, the main statement follows. We simultaneously show 1, 2 and 3 by induction on $|\varphi| + |\alpha^{\uparrow}| + |\alpha^{\downarrow}|$.

Let us see item 1. The base case is $\varphi = a$ for some $a \in \mathbb{A}$. By Harmony, $label(x) = label(x')$ and then $\mathcal{T}, x \models \varphi$ iff $\mathcal{T}', x' \models \varphi$. The boolean cases for φ are straightforward.

Suppose $\varphi = \langle \varepsilon = \alpha^{\uparrow}\alpha^{\downarrow} \rangle$. We show $\mathcal{T}, x \models \varphi \Rightarrow \mathcal{T}', x' \models \varphi$, so assume $\mathcal{T}, x \models \varphi$. Suppose there are $y, z \in \mathcal{T}$ and $n \leq \hat{s}$, $m \leq \hat{r} + n$ such that $y \xrightarrow{n} x$, $y \xrightarrow{m} z$, $(x,y) \in [\![\alpha^{\uparrow}]\!]^{\mathcal{T}}$, $(y,z) \in [\![\alpha^{\downarrow}]\!]^{\mathcal{T}}$ and $data(x) = data(z)$. By Zig, there are $y', z' \in T'$ such that $z Z^{\hat{k}-1}_{\hat{r}',\hat{s}'} z'$ for $\hat{r}' = \hat{r} + n - m$, $\hat{s}' = \hat{s} - n + m$, and $data(x') = data(z')$. By inductive hypothesis 2 and 3, $(x', y') \in [\![\alpha^{\uparrow}]\!]^{\mathcal{T}'}$ and $(y', z') \in [\![\alpha^{\downarrow}]\!]^{\mathcal{T}'}$. Hence $\mathcal{T}', x' \models \varphi$. The implication $\mathcal{T}', x' \models \varphi \Rightarrow \mathcal{T}, x \models \varphi$ is analogous. The cases $\varphi = \langle \varepsilon \neq \alpha^{\updownarrow} \rangle$, and $\varphi = \langle \varepsilon \odot \alpha^{\uparrow} \rangle$, $\varphi = \langle \varepsilon \odot \alpha^{\downarrow} \rangle$ ($\odot \in \{=, \neq\}$) and $\varphi = \langle \alpha \rangle$ (for α in up-down normal form) are shown in a similar way. The cases $\varphi = \langle \varepsilon \odot \varepsilon \rangle$ ($\odot \in \{=, \neq\}$) are trivial.

Let us now analyze item 2. Let $\alpha^{\uparrow} = [\psi]\uparrow^n$ ($n \geq 0$), and let

$$x_0, \ldots, x_n \in T \quad \text{and} \quad x'_0, \ldots, x'_n \in T'$$

be such that

$$\begin{aligned} y = x_0 \to x_1 \to \cdots \to x_n = x \qquad & \text{in } \mathcal{T}, \\ y' = x'_0 \to x'_1 \to \cdots \to x'_n = x' \qquad & \text{in } \mathcal{T}', \end{aligned}$$

and $x Z^{\hat{k}-1}_{\hat{r},\hat{s}} x'$. By Observation 3.14, we have $x_0 Z^{\hat{k}-1}_{\hat{r}',\hat{s}'} x'_0$, for $\hat{r}' = \hat{r}+n$, $\hat{s}' = \hat{s}-n$. Assume by contradiction that $(x', y') \notin [\![\alpha^{\uparrow}]\!]^{\mathcal{T}'}$. This necessarily means that $\mathcal{T}, x_0 \models \psi$ but $\mathcal{T}', x'_0 \not\models \psi$. But ψ is a subformula of $\alpha^{\uparrow}$, $\mathrm{nd}(\psi) \leq \hat{k} - 1$ and $\mathrm{nd}(\psi) \leq (\hat{r}', \hat{s}')$ and this contradicts inductive hypothesis 1.

Item 3 is shown in a similar way. Let $\alpha^{\downarrow} = \downarrow^m[\psi]$ ($m \geq 0$), and let

$$z_0, \ldots, z_m \in T \quad \text{and} \quad z'_0, \ldots, z'_m \in T'$$

be such that

$$\begin{aligned} y = z_0 \to z_1 \to \cdots \to z_m = z \qquad & \text{in } \mathcal{T}, \\ y' = z'_0 \to z'_1 \to \cdots \to z'_m = z' \qquad & \text{in } \mathcal{T}', \end{aligned}$$

and $z Z^{\hat{k}-1}_{\hat{r}',\hat{s}'} z'$. Assume by contradiction that $(y', z') \notin [\![\alpha^{\downarrow}]\!]^{\mathcal{T}'}$. This necessarily means that $\mathcal{T}, x_m \models \psi$ but $\mathcal{T}', x'_m \not\models \psi$. But ψ is a subformula of $\alpha^{\downarrow}$, $\mathrm{nd}(\psi) \leq \hat{k} - 1$ and $\mathrm{nd}(\psi) \leq (\hat{r}', \hat{s}')$ and this contradicts inductive hypothesis 1. $\square$

PROPOSITION A.5. $\mathcal{T}, u \equiv^{\updownarrow}_{r,s,k} \mathcal{T}', u'$ *implies* $\mathcal{T}, u \leftrightarrows^{\updownarrow}_{r,s,k} \mathcal{T}', u'$.

PROOF. Fix $u \in T$ and $u' \in T'$ such that $\mathcal{T}, u \equiv^{\updownarrow}_{r,s,k} \mathcal{T}', u'$. Define $(Z^{\hat{k}}_{\hat{r},\hat{s}})_{\hat{r}+\hat{s}\leq r+s, \hat{k}\leq k}$ by

$$x Z^{\hat{k}}_{\hat{r},\hat{s}} x' \quad \text{iff} \quad \mathcal{T}, x \equiv^{\updownarrow}_{\hat{r},\hat{s},\hat{k}} \mathcal{T}', x'.$$

We show that $Z^k_{r,s}$ is a (r,s,k)-bisimulation between $\mathcal{T}, u$ and $\mathcal{T}', u'$. By hypothesis, $u Z^k_{r,s} u'$. Now fix $\hat{r} + \hat{s} \leq r + s$, $\hat{k} \leq k$. By construction, $Z^{\hat{k}}_{\hat{r},\hat{s}}$ satisfies Harmony. Let us see that $Z^{\hat{k}}_{\hat{r},\hat{s}}$ satisfies Zig (the case for Zag is analogous). Suppose $x Z^{\hat{k}}_{\hat{r},\hat{s}} x'$,

$$\begin{aligned} y = x_0 \to x_1 \to \cdots \to v_n = x \qquad & \text{in } \mathcal{T}, \\ y = z_0 \to z_1 \to \cdots \to z_m = z \qquad & \text{in } \mathcal{T}, \end{aligned}$$

and $data(x) = data(z)$ (the case $data(x) \neq data(z)$ is shown in a similar way), where $m \leq \hat{r} + n$. Let $P \subseteq T'^2$ be defined by

$$P = \{(y', z') \mid y' \xrightarrow{n} x' \wedge y' \xrightarrow{m} z' \wedge data(x') = data(z')\}.$$

Since $\mathcal{T}, x \equiv^{\updownarrow}_{r,s,k} \mathcal{T}', x'$, $\mathrm{vd}(\langle \varepsilon = \uparrow^n \downarrow^m \rangle) \leq (r, s)$, $\mathrm{nd}(\langle \varepsilon = \uparrow^n \downarrow^m \rangle) = 0$, and $\mathcal{T}, x \models \langle \varepsilon = \uparrow^n \downarrow^m \rangle$, we conclude that $P \neq \emptyset$. We next show that there exists $(y', z') \in P$ such that

i. $y' = x'_0 \to x'_1 \to \cdots \to x'_n = x'$ in $\mathcal{T}'$

ii. $y' = z'_0 \to z'_1 \to \cdots \to z'_m = z'$ in $\mathcal{T}'$,

iii. $\mathcal{T}, x \equiv^{\updownarrow}_{\hat{r},\hat{s},\hat{k}-1} \mathcal{T}', x'$, and

iv. $\mathcal{T}, z \equiv^{\updownarrow}_{\hat{r}',\hat{s}',\hat{k}-1} \mathcal{T}', z'$, where $\hat{r}' = \hat{r} + n - m$, $\hat{s}' = \hat{s} - n + m$,

and hence, by inductive hypothesis, Zig is satisfied by $Z^{\hat{k}}_{\hat{r},\hat{s}}$. By way of contradiction, assume that for all $(y', z') \in P$ satisfying i and ii we have either

(a) $\mathcal{T}, x \not\equiv^{\updownarrow}_{\hat{r},\hat{s},\hat{k}-1} \mathcal{T}', x'$; or

(b) $\mathcal{T}, z \not\equiv^{\updownarrow}_{\hat{r}',\hat{s}',\hat{k}-1} \mathcal{T}', z'$ for $\hat{r}' = \hat{r} + n - m$, $\hat{s}' = \hat{s} - n + m$.

Fix $\top$ as any tautology such that $\mathrm{vd}(\top) = (0, 0)$, $\mathrm{nd}(\top) = 0$. For each $(y', z') \in P$ we define formulas, $\varphi_{y',z'}$ and $\psi_{y',z'}$, satisfying that $\mathrm{vd}(\varphi_{y',z'}) \leq (\hat{r}, \hat{s})$, $\mathrm{nd}(\varphi_{y',z'}) < \hat{k}$ and $\mathrm{vd}(\psi_{y',z'}) \leq (\hat{r}', \hat{s}')$, $\mathrm{nd}(\psi_{y',z'}) < \hat{k}$ as follows:

- Suppose (a) holds. Let $\varphi_{y',z'}$ be such that $\mathrm{vd}(\varphi_{v',w'}) \leq (\hat{r}, \hat{s})$, $\mathrm{nd}(\varphi_{v',w'}) < \hat{k}$, and such that $\mathcal{T}, x \models \varphi_{y',z'}$ but $\mathcal{T}', x' \not\models \varphi_{y',z'}$; and let $\psi_{v',w'} = \top$.
- Suppose (a) does not hold. Then (b) holds. Let $\psi_{y',z'}$ be such that $\mathrm{vd}(\psi_{y',z'}) \leq (\hat{r}', \hat{s}')$, $\mathrm{nd}(\psi_{y',z'}) < \hat{k}$ and such that $\mathcal{T}, z \models \psi_{y',z'}$ but $\mathcal{T}', z' \not\models \psi_{y',z'}$; and let $\varphi_{y',z'} = \top$.

Let

$$\Phi = \bigwedge_{(y',z') \in P} \varphi_{y',z'} \quad \text{and} \quad \Psi = \bigwedge_{(y',z') \in P} \psi_{y',z'}. \qquad (2)$$

Since $\mathrm{vd}(\varphi_{y',z'}) \leq (\hat{r}, \hat{s})$, $\mathrm{nd}(\varphi_{y',z'}) < \hat{k}$, by Proposition 3.12, there are finitely many non-equivalent formulas $\varphi_{y',z'}$. The same applies to formulas $\psi_{y',z'}$. Hence both infinite conjunctions in (2) are equivalent to finite ones, and therefore without loss of generality we may assume that Φ and Ψ are well-formed formulas.

Finally, let

$$\alpha^{\uparrow} = [\Phi]\uparrow^n \quad \text{and} \quad \alpha^{\downarrow} = \downarrow^m [\Psi].$$

By construction, $\mathrm{vd}(\alpha^{\uparrow}\alpha^{\downarrow}) \leq (\hat{r}, \hat{s})$, $\mathrm{nd}(\alpha^{\uparrow}\alpha^{\downarrow}) \leq \hat{k}$. Furthermore, $\mathcal{T}, x \models \langle \varepsilon = \alpha^{\uparrow}\alpha^{\downarrow} \rangle$ and $\mathcal{T}', x' \not\models \langle \varepsilon = \alpha^{\uparrow}\alpha^{\downarrow} \rangle$, but this contradicts the fact that $\mathcal{T}, x \equiv^{\updownarrow}_{\hat{r},\hat{s},\hat{k}} \mathcal{T}', x'$. □

Corollary 3.17. *$\leftrightarrows^{\updownarrow}_{r,s,k}$ has finite index.*

Proof. Immediate from Theorem 3.16 and Proposition 3.12. □

B. PROOFS OF SECTION 4

Theorem 4.1.

1. *Let $\dagger \in \{\downarrow, \updownarrow\}$. $\mathcal{T}, u \rightrightarrows^{\dagger} \mathcal{T}', u'$ implies $\mathcal{T}, u \Rrightarrow^{\dagger} \mathcal{T}', u'$. The converse holds when $\mathcal{T}'$ is finitely branching.*
2. *$\mathcal{T}, u \rightrightarrows^{\downarrow}_{\ell} \mathcal{T}', u'$ iff $\mathcal{T}, u \Rrightarrow^{\downarrow}_{\ell} \mathcal{T}', u'$.*
3. *$\mathcal{T}, u \rightrightarrows^{\updownarrow}_{r,s,k\cdot(r+s+2)} \mathcal{T}', u'$ implies $\mathcal{T}, u \Rrightarrow^{\updownarrow}_{r,s,k} \mathcal{T}', u'$.*
4. *$\mathcal{T}, u \Rrightarrow^{\updownarrow}_{r,s,k} \mathcal{T}', u'$ implies $\mathcal{T}, u \rightrightarrows^{\updownarrow}_{r,s,k} \mathcal{T}', u'$.*

Proof. The proofs are straightforward adaptations of the proofs of Propositions 3.8 and 3.9 and Propositions A.4 and A.5 respectively, and are ommitted here. In particular, for the 'if' part, in the adaptation of the proofs of Propositions 3.9 and A.5, the simulations are defined by

$$x Z_i x' \quad \text{iff} \quad \mathcal{T}, x \Rrightarrow^{\downarrow}_{i} \mathcal{T}', x$$
$$x Z^{\hat{k}}_{\hat{r},\hat{s}} x' \quad \text{iff} \quad \mathcal{T}, x \Rrightarrow^{\updownarrow}_{\hat{r},\hat{s},\hat{k}} \mathcal{T}', x$$

respectively, and the conditions (a) and (b) on page 5 become now

(a) $[\exists i \in \{0, \ldots, n\}\ \exists \varphi \in \mathrm{XPath}^{\downarrow+}_{=}]\ \mathrm{dd}(\varphi) \leq h - i \wedge \mathcal{T}, v_i \models \varphi \wedge \mathcal{T}', v'_i \not\models \varphi$; or

(b) $[\exists j \in \{0, \ldots, m\}\ \exists \varphi \in \mathrm{XPath}^{\downarrow+}_{=}]\ \mathrm{dd}(\varphi) \leq h - j \wedge \mathcal{T}, w_j \models \varphi \wedge \mathcal{T}', w'_j \not\models \varphi$,

and

(a) $[\exists i \in \{0, \ldots n\}\ \exists \varphi \in \mathrm{XPath}^{\updownarrow+}_{=}]\ \mathrm{vd}(\varphi) \leq (\hat{r} + i, \hat{s} - i) \wedge \mathrm{nd}(\varphi) \leq k - 1 \wedge \mathcal{T}, v_i \models \varphi \wedge \mathcal{T}', v'_i \not\models \varphi$; or

(b) $[\exists j \in \{0, \ldots m\}\ \exists \varphi \in \mathrm{XPath}^{\updownarrow+}_{=}]\ \mathrm{vd}(\varphi) \leq (\hat{r} + j', \hat{s} - j')$ for $j' = n - m + j \wedge \mathrm{nd}(\varphi) \leq k - 1 \wedge \mathcal{T}, w_j \models \varphi \wedge \mathcal{T}', w'_j \not\models \varphi$

respectively. □

Lemma B.1.

(1) $\{\mathcal{T}', u' \mid \mathcal{T}, u \rightrightarrows^{\downarrow}_{\ell} \mathcal{T}', u'\}$ is definable by an $\mathrm{XPath}^{\downarrow+}_{=}$-formula $\chi^{+}_{\ell,u,\mathcal{T}}$ of downward depth $\leq \ell$.

(2) $\{\mathcal{T}', u' \mid \mathcal{T}, u \rightrightarrows^{\updownarrow}_{r,s,k} \mathcal{T}', u'\}$ is definable by an $\mathrm{XPath}^{\updownarrow+}_{=}$-formula $\chi^{+}_{r,s,k,u,\mathcal{T}}$ of vertical depth $\leq (r, s)$ and nesting depth $\leq k$.

Proof. For item (2), let $sim^{\updownarrow}_{r,s,k}(\mathcal{T}, u) = \{\mathcal{T}', u' \mid \mathcal{T}, u \rightrightarrows^{\updownarrow}_{r,s,k} \mathcal{T}', u'\}$. Let $\Phi_{\mathcal{T}',u'}$ be the set of all positive formulas $\varphi \in \mathrm{XPath}^{\updownarrow+}_{=}$ of vertical depth at most (r, s) and nesting depth at most k so that $\mathcal{T}', u' \models \varphi$. Let Ψ be

$$\Psi = \bigvee_{\mathcal{T}',u' \in sim^{\updownarrow}_{r,s,k}(\mathcal{T},u)} \bigwedge \Phi_{\mathcal{T}',u'}.$$

Since every $\Phi_{\mathcal{T}',u'}$ is finite up to logical equivalence by Proposition 3.12, it follows that Ψ is a valid formula. We show that it defines $sim^{\updownarrow}_{r,s,k}(\mathcal{T}, u)$.

Let $\mathcal{T}', u' \in sim^{\updownarrow}_{r,s,k}(\mathcal{T}, u)$. Then, $\mathcal{T}', u' \models \bigwedge \Phi_{\mathcal{T}',u'}$ and thus $\mathcal{T}', u' \models \Psi$. If on the other hand $\mathcal{T}', u' \models \Psi$ we have that $\mathcal{T}', u' \models \bigwedge \Phi_{\mathcal{T}'',u''}$ for some $\mathcal{T}'', u'' \in sim^{\updownarrow}_{r,s,k}(\mathcal{T}, u)$ and then $\mathcal{T}', u' \equiv^{\updownarrow}_{r,s,k} \mathcal{T}'', u''$. By Theorem 3.16-3 we then have that $\mathcal{T}', u' \leftrightarrows^{\updownarrow}_{r,s,k} \mathcal{T}'', u''$, and in particular $\mathcal{T}'', u'' \rightrightarrows^{\updownarrow}_{r,s,k}$ □

$\mathcal{T}', u'$. Since $\mathcal{T}, u \rightrightarrows^{\updownarrow}_{r,s,k} \mathcal{T}'', u''$ and $\mathcal{T}'', u'' \rightrightarrows^{\updownarrow}_{r,s,k} \mathcal{T}', u'$, then $\mathcal{T}, u \rightrightarrows^{\updownarrow}_{r,s,k} \mathcal{T}', u'$ (by transitivity of $\rightrightarrows^{\updownarrow}_{r,s,k}$) and thus $\mathcal{T}', u' \in sim^{\updownarrow}_{r,s,k}(\mathcal{T}, u)$.

Item (1) is shown in a similar way, making use of Proposition 3.1 and Theorem 3.7-2. □

THEOREM 4.3.

1. *$\varphi \in XPath^{\downarrow}_{=}$ is $\rightrightarrows^{\downarrow}$-invariant [resp. $\rightrightarrows^{\downarrow}_{\ell}$] iff it is equivalent to a formula of $XPath^{\downarrow+}_{=}$ [resp. ℓ-$XPath^{\downarrow+}_{=}$].*
2. *$\varphi \in XPath^{\updownarrow}_{=}$ is $\rightrightarrows^{\updownarrow}$-invariant iff it is equivalent to a formula of $XPath^{\updownarrow+}_{=}$.*
3. *If $\varphi \in XPath^{\updownarrow}_{=}$ is $\rightrightarrows^{\updownarrow}_{r,s,k}$-invariant then it is equivalent to a formula of (r,s,k)-$XPath^{\updownarrow+}_{=}$.*
4. *If $\varphi \in XPath^{\updownarrow}_{=}$ is equivalent to a formula of (r,s,k)-$XPath^{\updownarrow+}_{=}$ then φ is $\rightrightarrows^{\updownarrow}_{r,s,k'}$-invariant, for $k' = k\cdot(r+s+2)$.*

PROOF. We start with item (1), for the case of $\rightrightarrows^{\downarrow}_{\ell}$. The 'if' part is straightforward from Theorem 4.1-2, and here we focus on the 'only if' part. Let φ be preserved under $\rightrightarrows^{\downarrow}_{\ell}$. Let $\{(\mathcal{T}_i, u_i)\}_{i\le n}$ be the set of all pointed models of φ modulo $\leftrightarrows^{\downarrow}_{\ell}$ (which is finite due to Theorem 3.7-2 together with Proposition 3.1). We claim that

$$\mathcal{T}, u \models \varphi \text{ iff } \mathcal{T}_i, u_i \rightrightarrows^{\downarrow}_{\ell} \mathcal{T}, u \text{ for some } i \le n. \qquad (3)$$

On the one hand, if $\mathcal{T}, u \models \varphi$ then there is $i \le n$ such that $\mathcal{T}_i, u_i \leftrightarrows^{\downarrow}_{\ell} \mathcal{T}, u$, and so $\mathcal{T}_i, u_i \rightrightarrows^{\downarrow}_{\ell} \mathcal{T}, u$. On the other hand, suppose $\mathcal{T}_i, u_i \rightrightarrows^{\downarrow}_{\ell} \mathcal{T}, u$. Since φ is preserved under $\rightrightarrows^{\downarrow}_{\ell}$ and $\mathcal{T}_i, u_i \models \varphi$, we conclude $\mathcal{T}, u \models \varphi$.

Let $\chi_{\ell,u_i,\mathcal{T}_i} \in \mathrm{XPath}^{\downarrow+}_{=}$, $\mathrm{dd}(\psi_i) \le \ell$, be as in Lemma B.1-(1). Using (3) one shows that $\bigvee_{i\le n} \chi_{\ell,u_i,\mathcal{T}_i} \equiv \varphi$.

For the case of $\rightrightarrows^{\downarrow}$ of item (1), the 'if' direction follows from Theorem 4.1-1. For the 'only if' direction, let φ be preserved under $\rightrightarrows^{\downarrow}$. It is easy to see that φ is preserved under $\rightrightarrows^{\downarrow}$ iff it is preserved under $\rightrightarrows^{\downarrow}_{\mathrm{dd}(\varphi)}$. We can then apply the same reasoning as before and the statement follows.

Item (3) follows the same argument as item (1) but this time using Corollary 3.17 and Lemma B.1-(2).

Item (4) is straightforward from Theorem 4.1-3.

Item (2) follows from items (3) and (4) and the observation that φ is preserved under $\rightrightarrows^{\updownarrow}$ iff it is preserved under $\rightrightarrows^{\updownarrow}_{r,s,k\cdot(r+s+2)}$ for $\mathrm{vd}(\varphi) = (r,s)$ and $\mathrm{nd}(\varphi) = k$. □

PROPOSITION B.2. *Any $\leftrightarrows^{\downarrow}$-invariant $\varphi(x) \in \mathrm{FO}(\sigma)$ over [finite] data-trees that is ℓ-local, is $\leftrightarrows^{\downarrow}_{\ell}$-invariant.*

PROOF. Let $\varphi(x)$ be ℓ-local and $\leftrightarrows^{\downarrow}$-invariant. Suppose $\mathcal{T}, u \leftrightarrows^{\downarrow}_{\ell} \mathcal{T}', u'$ and $\mathcal{T} \models \varphi(u)$. By ℓ-locality, $\mathcal{T}|_{\ell} u \models \varphi(u)$. Now

$$\begin{aligned} \mathcal{T}, u \leftrightarrows^{\downarrow}_{\ell} \mathcal{T}', u' &\text{ iff } (\mathcal{T}|_{\ell} u), u \leftrightarrows^{\downarrow}_{\ell} (\mathcal{T}'|_{\ell} u'), u' && \text{(Prop. 3.6)} \\ &\text{iff } (\mathcal{T}|_{\ell} u), u \leftrightarrows^{\downarrow} (\mathcal{T}'|_{\ell} u'), u'. && \text{(Prop. 3.5)} \end{aligned}$$

By $\leftrightarrows^{\downarrow}$-invariance, $\mathcal{T}'|_{\ell} u' \models \varphi(u')$ and by ℓ-locality again, $\mathcal{T}' \models \varphi(u')$. □

PROPOSITION B.3. *If $\varphi(x) \in \mathrm{FO}(\sigma)$ is $\leftrightarrows^{\downarrow}_{\ell}$-invariant over [finite] data-trees, then there is $\psi \in \ell$-$XPath^{\downarrow}_{=}$ such that $\mathrm{Tr}_x(\psi)$ is logically equivalent to φ over [finite] data-trees.*

PROOF. By Corollary 3.2, for every data tree $\mathcal{T}$ and $u \in T$ there is an ℓ-$XPath^{\downarrow}_{=}$ formula $\chi_{\ell,\mathcal{T},u}$ such that $\mathcal{T}, u \equiv^{\downarrow}_{\ell} \mathcal{T}', u'$ iff $\mathcal{T}', u' \models \chi_{\ell,\mathcal{T},u}$. Let

$$\psi = \bigvee_{\mathcal{T} \models \varphi(u)} \chi_{\ell,\mathcal{T},u}.$$

Since $\chi_{\ell,\mathcal{T},u} \in \ell$-$XPath^{\downarrow}_{=}$ and, by Proposition 3.1, $\equiv^{\downarrow}_{\ell}$ has finite index, it follows that ψ is equivalent to a finite disjunction.

We now show that $\varphi \equiv \mathrm{Tr}_x(\psi)$. Let us see that $\varphi \models \mathrm{Tr}_x(\psi)$. Suppose $\mathcal{T} \models \varphi(u)$. Since $\mathcal{T}, u \models \chi_{\ell,\mathcal{T},u}$, we have $\mathcal{T}, u \models \psi$ and so $\mathcal{T} \models \mathrm{Tr}_x(\psi)(u)$. Let us now see that $\mathrm{Tr}_x(\psi) \models \varphi$. Assume $\mathcal{T} \models \mathrm{Tr}_x(\psi)(u)$, and so $\mathcal{T}, u \models \psi$. Then there exists $\mathcal{T}', u'$ such that $\mathcal{T}' \models \varphi(u')$ and $\mathcal{T}, u \models \chi_{\ell,\mathcal{T}',u'}$. By the property of $\chi_{\ell,\mathcal{T}',u'}$, we have $\mathcal{T}, u \equiv^{\downarrow}_{\ell} \mathcal{T}', u'$ and since φ is $\leftrightarrows^{\downarrow}_{\ell}$-invariant (and hence $\equiv^{\downarrow}_{\ell}$-invariant by Theorem 3.7-2) we conclude $\mathcal{T} \models \varphi(u)$. □

C. PROOFS OF SECTION 5

THEOREM 5.1. *Let $\dagger \in \{\downarrow\downarrow_*, \updownarrow^*\}$.*

1. *$\mathcal{T}, u \leftrightarrows^{\dagger} \mathcal{T}', u'$ implies $\mathcal{T}, u \equiv^{\dagger} \mathcal{T}', u'$. The converse also holds when $\mathcal{T}'$ is finitely branching.*
2. *$\mathcal{T}, u \rightrightarrows^{\dagger} \mathcal{T}', u'$ implies $\mathcal{T}, u \Rrightarrow^{\dagger} \mathcal{T}', u'$. The converse also holds when $\mathcal{T}'$ is finitely branching.*

PROOF. The proof that $\mathcal{T}, u \leftrightarrows^{\downarrow} \mathcal{T}', u' \Rightarrow \mathcal{T}, u \equiv^{\downarrow\downarrow_*} \mathcal{T}', u'$ follows from a simple adaptation of Proposition 3.8 to the logic $\mathrm{XPath}^{\downarrow\downarrow_*}_{=}$ and Lemma A.1. The fact that for finitely branching, $\mathcal{T}, u \equiv^{\downarrow\downarrow_*} \mathcal{T}', u' \Rightarrow \mathcal{T}, u \leftrightarrows^{\downarrow} \mathcal{T}', u'$ is straightforward from Theorem 3.7-1 since $\equiv^{\downarrow\downarrow_*} \subseteq \equiv^{\downarrow}$.

The cases for $\mathrm{XPath}^{\updownarrow^*}_{=}$, $\mathrm{XPath}^{\downarrow\downarrow_*}_{=}$ and $\mathrm{XPath}^{\updownarrow^{*}+}_{=}$ are analogous. □

D. PROOFS OF SECTION 6

PROPOSITION 6.1. *For any $XPath^{\updownarrow}_{=}$- [resp. $XPath^{\downarrow}_{=}$-] path expression α there is an equivalent $XPath^{\updownarrow}_{=}$- [resp. $XPath^{\downarrow}_{=}$-] path expression α' in simple normal form. Further, α' can be computed in polynomial time from α.*

PROOF. The translation is straightforward, given the following equivalences:

$$\begin{aligned} \varepsilon &\equiv [\top] \\ \alpha &\equiv [\top]\alpha \equiv \alpha[\top] \\ \alpha[\varphi][\psi]\beta &\equiv \alpha[\varphi \wedge \psi]\beta \end{aligned}$$

where $\top$ denotes any fixed tautology, for example $a \vee \neg a$, for some $a \in \mathcal{A}$. □

LEMMA 6.5. *The $\mathrm{FO}(\sigma)$-formula*

$$(\exists x)\ P_a(x)$$

is $\leftrightarrows^{\updownarrow}$-invariant though not logically equivalent over [finite] data-trees to any $XPath^{\updownarrow}_{=}$-formula.

PROOF. Let $\varphi(x)$ be the $\mathrm{FO}(\sigma)$-formula for *there is a node labeled a in the tree*, i.e.,

$$\varphi(x) = (\exists y)\ P_a(y).$$

We prove that φ is $\leftrightarrows^{\updownarrow}$-invariant over [finite] data-trees, though it is not logically equivalent over [finite] data-trees to any XPath$^{\updownarrow}_{=}$-formula.

To see that φ is $\leftrightarrows^{\updownarrow}$-invariant over [finite] data-trees, take $\mathcal{T}, u$ and $\mathcal{T}', u'$ such that $\mathcal{T}, u \leftrightarrows^{\updownarrow} \mathcal{T}', u'$ and $\mathcal{T} \models \varphi(u)$. Furthermore, suppose that $\mathcal{T}, u \models \uparrow^m\downarrow^n a$ for adequate n and m. By Theorem 3.16, $\mathcal{T}', u' \models \uparrow^n\downarrow^m a$ and so $\mathcal{T}' \models \varphi(u')$.

Assume by contradiction that there is $\psi \in$ XPath$^{\updownarrow}_{=}$ such that $\mathcal{T}, u \models \psi$ iff $\mathcal{T} \models \varphi(u)$ for all data-tree $\mathcal{T}$ and $u \in T$. Suppose $\mathrm{vd}(\psi) = (r, s)$ and $\mathrm{nd}(\psi) = k$. Let $\mathcal{T}$ be a data tree formed by a chain of length $r + 1$ starting from the root u with all its nodes containing a label b except the leave, which has label a (the data values are irrelevant). By Proposition 3.15 we have $\mathcal{T}, u \leftrightarrows^{\updownarrow}_{r,s,k} (\mathcal{T}|^s_r u), u$. Since $\mathcal{T}, u \models \psi$, by Theorem 3.16, we have $(\mathcal{T}|^s_r u), u \models \psi$, and so $\mathcal{T}|^s_r u \models \varphi(u)$. This last fact is a contradiction because no node of $\mathcal{T}|^s_r u$ is labeled with a. □

Proposition 6.6. *Let $k' = k\cdot(r+s+2)$. If $\varphi(x) \in \mathrm{FO}(\sigma)$ is $\leftrightarrows^{\updownarrow}_{r,s,k'}$-invariant over [finite] data-trees, then there is $\psi \in (r, s, k)$-XPath$^{\updownarrow}_{=}$ such that $\mathrm{Tr}_x(\psi)$ is logically equivalent to φ over [finite] data-trees.*

Proof. By Corollary 3.13, for every data tree $\mathcal{T}$ and $u \in T$ there is an (r, s, k)-XPath$^{\updownarrow}_{=}$ formula $\chi_{r,s,k,\mathcal{T},u}$ such that $\mathcal{T}, u \equiv^{\updownarrow}_{r,s,k} \mathcal{T}', u'$ iff $\mathcal{T}', u' \models \chi_{r,s,k,\mathcal{T},u}$. Let

$$\psi = \bigvee_{\mathcal{T} \models \varphi(u)} \chi_{r,s,k,\mathcal{T},u}.$$

As $\chi_{r,s,k,\mathcal{T},u} \in (r, s, k)$-XPath$^{\updownarrow}_{=}$ and, by Proposition 3.12, $\equiv^{\updownarrow}_{r,s,k}$ has finite index, it follows that ψ is equivalent to a finite disjunction. The proof that $\varphi(x) \equiv \mathrm{Tr}_x(\psi)$ is similar to Proposition B.3, as we show next. Let us see that $\varphi \models \mathrm{Tr}_x(\psi)$. Suppose $\mathcal{T} \models \varphi(u)$. Since $\mathcal{T}, u \models \chi_{r,s,k,\mathcal{T},a}$, we have $\mathcal{T}, u \models \psi$ and so $\mathcal{T} \models \mathrm{Tr}_x(\psi)(u)$. Let us see that $\mathrm{Tr}_x(\psi) \models \varphi$. Assume $\mathcal{T} \models \mathrm{Tr}_x(\psi)(u)$, and so $\mathcal{T}, u \models \psi$. Then there exists $\mathcal{T}', u'$ such that $\mathcal{T}' \models \varphi(u')$ and $\mathcal{T}, u \models \chi_{r,s,k,\mathcal{T}',u'}$. By the property of $\chi_{r,s,k,\mathcal{T}',u'}$, we have $\mathcal{T}, u \equiv^{\updownarrow}_{r,s,k} \mathcal{T}', u'$ and since φ is $\leftrightarrows^{\downarrow}_{r,s,k\cdot(r+s+2)}$-invariant (and hence $\equiv^{\downarrow}_{r,s,k}$-invariant by Theorem 3.16-2) we conclude $\mathcal{T} \models \varphi(u)$. □

Lemma 6.7. *The* FO(σ)*-formula*

$$(\exists y, z)\ [y \approx z \wedge P_a(y) \wedge P_b(z)]$$

is $\leftrightarrows^{\updownarrow}$-invariant though not logically equivalent over [finite] data-trees to any XPath$^{\updownarrow E}_{=}$-formula.

Proof. Let $\varphi(x)$ be the FO(σ)-formula for *there are two nodes with same data value and labels a and b respectively*, i.e.,

$$\varphi(x) = (\exists y, z)\ [y \approx z \wedge P_a(y) \wedge P_b(z)].$$

We show that φ cannot be expressed in XPath$^{\downarrow,\uparrow,E}_{=}$. Suppose, by means of contradition, that there is a formula $\psi \in$ XPath$^{\downarrow,\uparrow,E}_{=}$ expressing φ, with $\mathrm{vd}(\psi) = (r, s)$ ($\mathrm{vd}(\cdot)$ for XPath$^{\downarrow,\uparrow,E}_{=}$ is defined as in Table 2 plus the clause $\mathrm{vd}(E\varphi) = \mathrm{vd}(\varphi)$). Let $n = r + s$, and let $\mathcal{T}$ be the chain-like data-tree

$$u_0 \to u_1 \to \cdots \to u_n$$

such that $label(u_0) = a$, $label(u_n) = b$, $label(u_i) = c$ for $i \in \{1, \ldots n-1\}$ and $data(u_i) = i$ for $i \in \{0, \ldots, n\}$. Let $\mathcal{T}'$ be the chain-like data-tree

$$u'_0 \to u'_1 \to \cdots \to u'_n$$

such that $label(u'_i) = label(u_i)$ for $i \in \{0, \ldots n\}$, $data(u'_i) = data(u_i)$ for $i \in \{0, \ldots, n-1\}$ and $data(u'_n) = 0$. Note that $\mathcal{T} \not\models \varphi(u_0)$ and $\mathcal{T}' \models \varphi(u'_0)$. However, one can show that for all $i \in \{0, \ldots, n\}$ we have $\mathcal{T}, u_i \models \psi$ iff $\mathcal{T}', u'_i \models \psi$. Hence, ψ does not express φ and thus φ is not expressible in XPath$^{\downarrow,\uparrow,E}_{=}$. □

E. PROOFS OF SECTION 7

Theorem 7.2. *For all $\ell, k \geq 0$, $i \geq 1$,*

$$\equiv^{\downarrow}_{\ell,0} \supsetneq \equiv^{\downarrow}_{\ell,1} \supsetneq \cdots \supsetneq \equiv^{\downarrow}_{\ell,\ell} = \equiv^{\downarrow}_{\ell,\ell+i}, \text{ and}$$
$$\equiv^{\downarrow}_{\ell,k} \supsetneq \equiv^{\downarrow}_{\ell+i,k}.$$

Proof. Consider the data trees defined in Figure 12 for every k. Note that $\equiv^{\downarrow}_{\ell,k+1} \subseteq \equiv^{\downarrow}_{\ell,k}$ and $\equiv^{\downarrow}_{\ell+1,k} \subseteq \equiv^{\downarrow}_{\ell,k}$ by definition. We show that $\equiv^{\downarrow}_{\ell,k} \neq \equiv^{\downarrow}_{\ell,k+1}$ for all $\ell \geq k+1$. For this purpose, we show that $\mathcal{T}^1_k, x^1_k \equiv^{\downarrow}_{k+1,k} \mathcal{T}'^1_k, x'^1_k$ but $\mathcal{T}^1_k, x^1_k \not\equiv^{\downarrow}_{k+1,k+1} \mathcal{T}'^1_k, x'^1_k$.

The fact that $\mathcal{T}^1_k, x^1_k \not\equiv^{\downarrow}_{k+1,k+1} \mathcal{T}'^1_k, x'^1_k$ comes from the fact that the property "there is a path of length $k+1$ ending with a label a whose every pair of consecutive nodes have distinct data value" is definable with the following formula φ_{k+1} of depth $k+1$ and nesting depth $k+1$,

$$\varphi_1 = \langle \varepsilon \neq \downarrow[a] \rangle$$
$$\varphi_{i+1} = \langle \varepsilon \neq \downarrow[\varphi_i] \rangle \quad \text{for } i > 0.$$

Since $\mathcal{T}^1_k, x^1_k \models \varphi_{k+1}$ but $\mathcal{T}'^1_k, x'^1_k \not\models \varphi_{k+1}$, it follows that $\mathcal{T}^1_k, x^1_k \not\equiv^{\downarrow}_{k+1,k+1} \mathcal{T}'^1_k, x'^1_k$.

To show $\mathcal{T}^1_k, x^1_k \equiv^{\downarrow}_{k+1,k} \mathcal{T}'^1_k, x'^1_k$ we actually use Proposition 7.1 and show $\mathcal{T}^1_k, x^1_k \leftrightarrows^{\downarrow}_{k+1,k} \mathcal{T}'^1_k, x'^1_k$. Note that $\mathcal{T}^1_k$ and $\mathcal{T}^2_k$ (resp. $\mathcal{T}'^1_k$ and $\mathcal{T}'^2_k$) are equal modulo renaming of data values, so we are also showing that the roots of any two data trees with subindex k are $(k+1, k)$-bisimilar.

Observation E.1. *Note that the set of immediate subtrees of the roots of $\mathcal{T}^1_k, \mathcal{T}'^1_k, \mathcal{T}^2_k, \mathcal{T}'^2_k$ are the same as those of $\mathcal{T}'^1_k, \mathcal{T}^2_k, \mathcal{T}'^2_k$ (and of $\mathcal{T}^1_k, \mathcal{T}'^1_k, \mathcal{T}'^2_k$) by construction.*

We now show $\mathcal{T}^1_k, x^1_k \leftrightarrows^{\downarrow}_{k+1,k} \mathcal{T}'^1_k, x'^1_k$. For every $j \leq k+1$, $t \leq k$, let $Z_{j,t}$ be the set of all pairs $(x, y) \in T^1_k \times T'^1_k$ so that x and y are some $x^i_{k'}$ or $x'^i_{k'}$ for $i \in \{1, 2\}$ and $k' \geq t$.[2] Observe that

$$Z_{j+1,t} \subseteq Z_{j,t} \quad \text{for all } j, t \leq k. \tag{4}$$

We show that $(Z_{j,t})_{j \leq k+1, t \leq k}$ verify the bisimulation conditions. We proceed by induction on $j + t$. The base case, $j = t = 0$, is trivial. The case $l > 0$, $t = 0$ is also straightforward.

Suppose then that $t > 0$. Let $(u, u') \in Z_{j,t}$. Again, Harmony is met since $Z_{l,t}$ relates only nodes with label a. Let us suppose that u is some $x^1_{t'}$ and u' is $x'^1_{t'}$ for some $t' \leq t$, the other cases being similar or simpler.

[2]Note that $x^i_{k'}$ or $x'^i_{k'}$ do not necessarily uniquely identify one node, but many possible. The intended meaning is that x, y can be *any* of them.

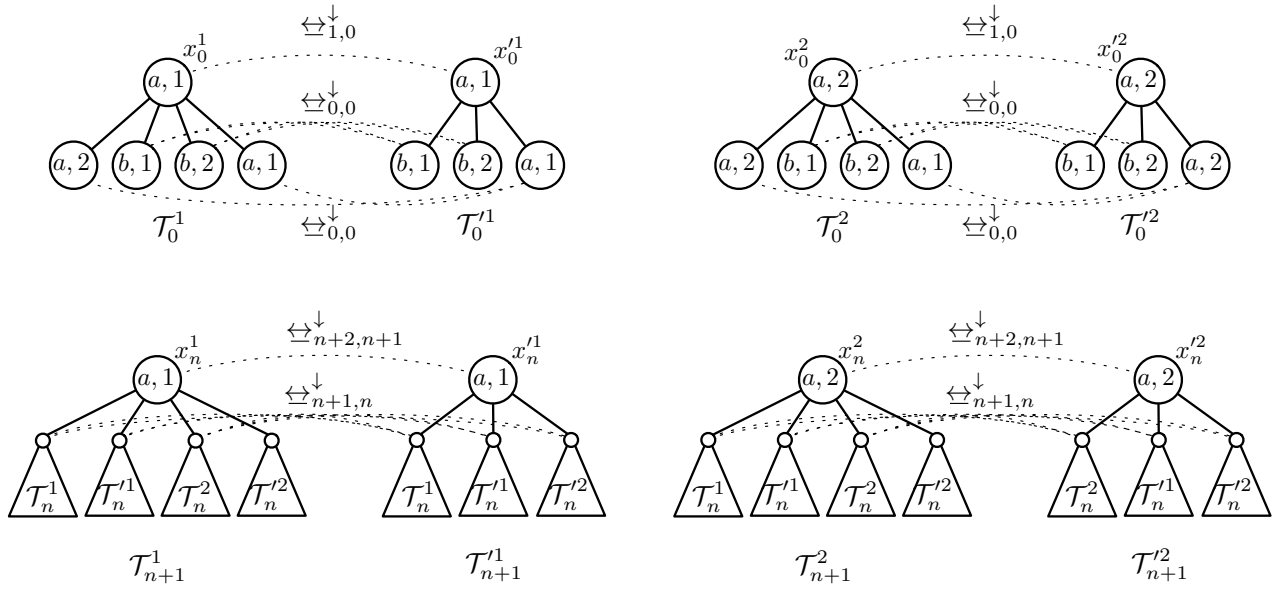

Figure 12: Definition of data trees $\mathcal{T}^i_n$, $\mathcal{T}'^i_n$ $(n \geq 0, i \in \{1,2\})$ for proof of Theorem 7.2.

Let us now show Zig. Let v, w be so that $x^1_{t'} \xrightarrow{n} v$ and $x'^1_{t'} \xrightarrow{m} w$ with $n, m \leq j$.

- If v is inside the subtree $\mathcal{T}^2_{t'-1}$ of $\mathcal{T}^1_{t'}$, but it is not $x^2_{t'-1}$, then we choose v' as the corresponding[3] node inside the subtree $\mathcal{T}^1_{t'-1}$ of $\mathcal{T}'^1_{t'}$. Note that $data(v) = data(v')$ by Observation E.1. Further, since every node of $\mathcal{T}^1_{t'-1}$ is in a $Z_{j,t-1}$-relation with the corresponding node in $\mathcal{T}^2_{t'-1}$ by construction of $Z_{j,t-1}$, it follows that $(\xrightarrow{i} v) Z_{j,t-1} (\xrightarrow{i} v')$ for all $i \leq n$. Thus, by (4), $(\xrightarrow{i} v) Z_{j-n+i,t-1} (\xrightarrow{i} v')$ for all $i \leq n$.

- If, on the other hand, v is $x^2_{t'-1}$, we choose v' as the root of $\mathcal{T}'^2_{t'-1}$, $x'^2_{t'-1}$. Again, we have that $data(v') = data(v)$ and by construction that $v Z_{j,t-1} v'$. Thus, by (4), $v Z_{j-1,t-1} v'$.

- Finally, if v falls outside $\mathcal{T}'^2_{t'-1}$, we choose v' as the same node in $\mathcal{T}'^1_{t'}$, where of course we will have that $data(v) = data(v')$ and that $(\xrightarrow{i} v) Z_{j,t-1} (\xrightarrow{i} v')$ for all $i \leq n$. Thus, by (4), $(\xrightarrow{i} v) Z_{j-n+i,t-1} (\xrightarrow{i} v')$ for all $i \leq n$.

We do the same with w and w'. Since in every case we can reach a node with the same data value and so that the corresponding nodes in the path are $Z_{j,t-1}$-related, it follows that the Zig condition is satisfied. The Zag condition is only easier, and hence we conclude that $\mathcal{T}^1_k, x \leftrightarrows^{\downarrow}_{k+1,k} \mathcal{T}'^1_k, x'$ for every k.

We therefore have that $\equiv^{\downarrow}_{\ell,k+1} \subsetneq \equiv^{\downarrow}_{\ell,k}$ for all $\ell \geq k+1$.

The fact that $\equiv^{\downarrow}_{\ell+1,k} \subsetneq \equiv^{\downarrow}_{\ell,k}$ is of course trivial, formulas of depth $\ell+1$ can express "the tree has at least depth $\ell+1$", which cannot be expressed by formulas of depth ℓ.

[3]Remember that $\mathcal{T}^1_{t'-1}$ and $\mathcal{T}^2_{t'-1}$ are isomorphic modulo a renaming of data values, so by *corresponding* we mean the node in the same position in the tree

It remains to show that $\equiv^{\downarrow}_{\ell,k} = \equiv^{\downarrow}_{\ell,k+1}$ for all $\ell \leq k$. To show this, we prove $\mathcal{T}, x \leftrightarrows^{\downarrow}_{\ell,k+1} \mathcal{T}', x'$ for every $\mathcal{T}, \mathcal{T}'$ so that $\mathcal{T}, x \leftrightarrows^{\downarrow}_{\ell,k} \mathcal{T}', x'$. We prove it by induction on $\ell + k$. The base case is easy.

For the inductive case, let $Z_{j,t} = \leftrightarrows^{\downarrow}_{j,t}$ for all $j \leq \ell, t \leq k$. Hence, $(Z_{j,t})_{j \leq \ell, t \leq k}$ verify the bisimulation conditions. Let $Z_{\ell,k+1} = \{(x, x')\}$. We show that $Z_{\ell,k+1}$ together with $(Z_{j,t})_{j \leq \ell, t \leq k}$ verifies the bisimulation conditions. Harmony follows from $x Z_{\ell,k} x'$. We show Zig since Zag is equivalent. Suppose $x \xrightarrow{n} v$, $x \xrightarrow{m} w$ with $n, m \leq \ell$. Then, since $Z_{\ell,k}$ verifies Zig, there are $x' \xrightarrow{n} v'$, $x' \xrightarrow{m} w'$ where

(1) $data(v) = data(w')$ iff $data(v') = data(w')$,

(2) $(\xrightarrow{i} v) Z_{\ell-n+i,k-1} (\xrightarrow{i} v')$ for all $i \in \{0, \ldots, n-1\}$, and

(3) $(\xrightarrow{i} w) Z_{\ell-m+i,k-1} (\xrightarrow{i} w')$ for all $i \in \{0, \ldots, m-1\}$.

Since $\ell \leq k$, then $\ell - n + i \leq k-1$. Further, $\ell - n + i + k < \ell + k$, which means that we can apply the inductive hypothesis. Hence, by inductive hypothesis, $\mathcal{T}, (\xrightarrow{i} v) \leftrightarrows^{\downarrow}_{\ell-n+i,k} \mathcal{T}', (\xrightarrow{i} v')$ and thus $(\xrightarrow{i} v) Z_{\ell-n+i,k} (\xrightarrow{i} v')$. By an indentical reasoning, $\mathcal{T}, (\xrightarrow{i} w) \leftrightarrows^{\downarrow}_{\ell-n+i,k} \mathcal{T}', (\xrightarrow{i} w')$ and thus $(\xrightarrow{i} w) Z_{\ell-n+i,k} (\xrightarrow{i} w')$. Thus, the Zig condition for $\leftrightarrows^{\downarrow}_{\ell,k+1}$ is verified. The Zag condition holds by symmetry. □

With respect to vertical XPath, note that since $\equiv^{\updownarrow}_{r,s,k} \subseteq \equiv^{\updownarrow}_{r',s',k'}$ for all $(r,s,k) \leq (r',s',k')$, as a consequence of Proposition 3.11 we obtain that for every r, s, k with $r+s \geq 2$ there is some $k' > k$ so that $\equiv^{\updownarrow}_{r,s,k} \supsetneq \equiv^{\updownarrow}_{r,s,k'}$. In fact, we conjecture that $\equiv^{\updownarrow}_{r,s,k} \supsetneq \equiv^{\updownarrow}_{r,s,k+1}$ for every k. We argue that this can be proven through the models $(\mathcal{T}_n)_n$ in the proof of Proposition 3.11, by showing that $\mathcal{T}_k, x_{r',s'} \equiv^{\updownarrow}_{r,s,k} \mathcal{T}_{k+1}, x_{r',s'}$ but $\mathcal{T}_k, x_{r',s'} \not\equiv^{\updownarrow}_{r,s,k+1} \mathcal{T}_{k+1}, , x_{r',s'}$ for every $(r,s) \geq (r',s')$. The fact that $\equiv^{\updownarrow}_{r,s,k} \supsetneq \equiv^{\updownarrow}_{r+1,s,k}$ and $\equiv^{\updownarrow}_{r,s,k} \supsetneq \equiv^{\updownarrow}_{r,s+1,k}$ are straightforward. We then obtain the following.

CLAIM E.2. $\equiv^{\updownarrow}_{r,s,k} \supsetneq \equiv^{\updownarrow}_{r',s',k'}$ *for all* $(r,s,k) < (r',s',k')$, $r+s \geq 2$.