# Characterizations for $\text{XPath}_{\mathcal{R}}(\downarrow)$

Nicolás González[1] and Sergio Abriola[1,2(✉)]

[1] University of Buenos Aires, Buenos Aires, Argentina
sabriola@dc.uba.ar
[2] ICC-CONICET, Buenos Aires, Argentina

**Abstract.** Over the semantic universe of trees augmented with arbitrary sets of relations between nodes, we study model-theoretic properties of the extension $\text{XPath}_{\mathcal{R}}(\downarrow)$ of the downward fragment of XPath, equipped with a finite set $\mathcal{R}$ of relation symbols. We introduce an adequate notion of bisimulation, dependant on the set of relations $\mathcal{R}$ in consideration, and show a characterization result in the style of Hennessy-Milner's, relating bisimulation and logical equivalence and showing that both coincide over finitely branching $\mathcal{R}$-trees. Furthermore, we also give a van Benthem-like theorem characterizing each $\text{XPath}_{\mathcal{R}}(\downarrow)$ as the fragment of first-order logic (over an adequate signature) with one free variable that is $\mathcal{R}$-bisimulation-invariant. Finally, we show that our results are also valid when applied to universes of trees with some fixed semantics for the symbols of $\mathcal{R}$. This contains in particular the case of $\text{XPath}_{=}(\downarrow)$ over data trees.

**Keywords:** XPath · Bisimulation · Characterization · Data logics

## 1 Introduction

XPath is the most widely used query language for XML documents; it is an open standard and constitutes a World Wide Web Consortium (W3C) Recommendation [1]. XPath has syntactic operators to navigate the tree using accessibility relations such as 'child', 'parent', 'sibling', et cetera, and can make tests on intermediate nodes. Core-XPath [2] is the fragment of XPath 1.0 containing only the navigational behavior of XPath. Core-XPath can express properties on nodes with respect to the underlying tree structure of the XML document, such as 'nodes with label B', or 'nodes that have both a child with label A and a grandchild with label B'. It can also express properties on paths along the tree such as 'the ending node is the grandchild of the starting node', or 'the initial node has label A and has a child with A, and the ending node is the grandparent of the starting node'. The first type of formulas are evaluated on individual nodes and are called *node expressions*, while the formulas of the second type are evaluated on pairs of nodes and are called *path expressions*. However, Core-XPath cannot express conditions on the actual data contained in the attributes, such as with a node expression saying 'this node has two children with different data values', or


A. Silva et al. (Eds.): WoLLIC 2021, LNCS 13038, pp. 319–336, 2021.
https://doi.org/10.1007/978-3-030-88853-4_20

'the value of this node coincides with the value of some descendant'. In contrast, Core-Data-XPath [3] (which we here call simply $\text{XPath}_=$) can perform these data comparisons. Indeed, $\text{XPath}_=$ is the extension of Core-XPath with the addition of (in)equality tests between attributes of elements in an XML document.

In the paper [4], the expressive power of fragments of $\text{XPath}_=$ was studied, from a logical and model-theoretical point of view, when the set of navegational axes was taken among $\downarrow, \uparrow$, and the reflexive-transitive closure of those axes. In that work, the semantic universe of study was that of *data trees*, whose nodes have a single label taken from a finite alphabet and a single data value from an infinite domain. A focus of study in that work was that of bisimulation, which is a classic tool of modal logics, used to determine equivalence between relational models. A node $x$ of a data tree $T$ and a node $x'$ of a data tree $T'$ are said to be *bisimilar* if they satisfy some special (depending of the studied fragment) back-and-forth conditions over the structure of the data tree. In [4], suitable notions of bisimulation were deviced for the $\text{XPath}_=$ fragments under considerations. Then, showing a characterization result in the style of the Hennessy-Milner's theorem for Basic Modal Logic[6], it was proven that if $x$ and $x'$ are bisimilar then they satisfy exactly the same node expressions, and that the converse is also true for trees whose every node only has a finite number of children. Hence, bisimulation coincides with logical equivalence, i.e., with *indistinguishability by means of node expressions*. The paper [4] also stated and proved theorems in the style of van Benthem's for Basic Modal Logic [7], but in the context of $\text{XPath}_=$. One of these theorems states that the downward fragment of $\text{XPath}_=$ coincides with the bisimulation-invariant fragment of first-order logic with one free variable (over the adequate signature). For the case of the vertical fragment of $\text{XPath}_=$, this characterization fails, but a weaker result is proved instead.

In [5], the study of bisimulation for $\text{XPath}_=$ was expanded in order to encompass bisimulation notions over two-pointed data trees (i.e. a data tree and two specified nodes), giving a bisimulation notion for the downward and the vertical fragment of $\text{XPath}_=$ over two-pointed data trees, and proving the corresponding characterization results between logical equivalence and bisimulation. In this way, the paper expanded the results of [4] from the domain of node expressions to that of path expressions.

In the current work we focus our study in a family of generalizations of $\text{XPath}_=(\downarrow)$, which includes not only the capability of comparing the end nodes of two paths by data (in)equality, but also checking for other types of arbitrary $n$-ary predicates over nodes at the end nodes of $n$ paths. Given a fixed set $\mathcal{R}$ of relation symbols with their arity, we generalize the concept of data trees to encompass arbitrary relations between nodes, and study the logic $\text{XPath}_{\mathcal{R}}(\downarrow)$ over this universe. We give a general suitable bisimulation notion for these $\mathcal{R}$ fragments, and show a Hennessy-Milner-like characterization result connecting this notion with that of logical equivalence. Furthermore, we provide a theorem in the style of van Benthem's result for Basic Modal Logic, thus characterizing these logics as fragments of first-order logic whose formulas are invariant under this new notion of bisimulation. While we initially state these results for the case of the full universe with no semantic restrictions, in the end we show that restricting ourselves to some universes of data trees preserves our results.

## 2 Preliminaries

A *data tree* is a directed tree whose nodes have a single label from a finite alphabet and a single data value from a (possibly infinite) data domain. XPath$_=(\downarrow)$ is a logic that can express properties about these structures, for instance, we can say if a node has or not certain label or if a node is a child of another via the child axis $\downarrow$. The most important capabilities of this logic lie in its *data tests* $\langle \alpha = \beta \rangle$ or $\langle \alpha \neq \beta \rangle$, which can compare the data values of two nodes. More precisely, a node satisfying $\langle \alpha = \beta \rangle$ (resp. $\langle \alpha \neq \beta \rangle$) means that there are paths from it such that the first one satisfies $\alpha$, the second one satisfies $\beta$ and their final points have equal (resp. non-equal) data value. Having the same data value can be thought of as a binary equivalence relation between nodes, so a natural question that arises is the possibility of extending the types of comparisons that can be made at the end of paths.

Fixed $\mathcal{R}$, XPath$_{\mathcal{R}}(\downarrow)$ is an extension of XPath$_=(\downarrow)$ in the sense that now it can allow relations with arbitrary arity (not necessarily binary) between final points of paths from a certain node of a tree. Here, formulas of the type $\langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle_r$ express this type of operation, where $r$ represents a particular relation symbol and $\mathcal{R}$ is the set of such symbols. In this context, the data test $\varphi := \langle \alpha = \beta \rangle$ can be re-expressed as $\langle \alpha, \beta \rangle_{=_d}$ where the sub index $=_d$ can be interpreted over a data tree as the equivalence relation of having same data value. Similarly, a label test $\varphi := a$ can be re-expressed as $\langle \varepsilon \rangle_a$, where $a$ is an unary relation symbol.

Initially, we will consider that the universe of the models for XPath$_{\mathcal{R}}(\downarrow)$ is still that of trees, but extended with an *arbitrary* relation over nodes for each $r \in \mathcal{R}$ (with its respective arity). We call these models $\mathcal{R}$-trees.

**Definition 1.** *Let $\mathcal{R}$ be a finite and non-empty set of relational symbols with arities given by the function $\mathcal{A} : \mathcal{R} \to \mathbb{N}_{\geq 1}$. The formulas of XPath$_{\mathcal{R}}(\downarrow)$ are defined by the grammar below:*

$$\varphi, \psi = \varphi \wedge \psi \mid \neg\varphi \mid \langle \alpha_1, \alpha_2, \ldots, \alpha_{\mathcal{A}(r)} \rangle_r \mid \langle \alpha_1, \alpha_2, \ldots, \alpha_{\mathcal{A}(r)} \rangle_{\overline{r}} \qquad r \in \mathcal{R}$$
$$\alpha, \beta = \varepsilon \mid \downarrow \mid \alpha\beta \mid \alpha \cup \beta \mid [\varphi]$$

*The first row generates the* **node expressions**, *and the second one, the* **path expressions**: $\langle \alpha_1, \alpha_2, \ldots, \alpha_{\mathcal{A}(r)} \rangle_r$ *and* $\langle \alpha_1, \alpha_2, \ldots, \alpha_{\mathcal{A}(r)} \rangle_{\overline{r}}$ *are called* path tests *where $\alpha_i$ is a path expression for all $i \in \{1, 2, \ldots, \mathcal{A}(r)\}$; $\alpha\beta$ and $\alpha \cup \beta$ are respectively the* concatenation *and* union *between $\alpha$ and $\beta$; $[\varphi]$ are* node tests *(as part of a path expression) where $\varphi$ is a node expression; and the symbols $\varepsilon$ and $\downarrow$ are the* self *and* child *axes, respectively.*

*As usual we use $\varphi \vee \psi$ as a shorthand for $\neg(\neg\varphi \wedge \neg\psi)$. By $\overline{\boldsymbol{\mathcal{R}}}$ we indicate the set of the complement symbols $\overline{r}$ with $r \in \mathcal{R}$ and extend $\mathcal{A}$ to $\mathcal{R} \cup \overline{\mathcal{R}}$ as* $\boldsymbol{\mathcal{A}(\overline{r}) := \mathcal{A}(r)}$ *for $r \in \mathcal{R}$.*

From now on, we consider a fixed $\mathcal{R} \neq \emptyset$, a family of predicate symbols as in Definition 1.

**Definition 2.** *An* $\boldsymbol{\mathcal{R}}$*-tree* $\mathcal{T} = \langle T, E_\downarrow, \{R_r\}_{r\in\mathcal{R}}\rangle$ *is a set* $T$ *with a binary relation* $E_\downarrow \subseteq T^2$ *such that* $\langle T, E_\downarrow\rangle$ *is a rooted tree, and with a family of relations* $\{R_r\}_{r\in\mathcal{R}}$*, where* $R_r \subseteq T^{\mathcal{A}(r)}$*. We use* $R_{\overline{r}}$ *to abbreviate the complement of* $R_r$*, that is,* $\boldsymbol{R_{\overline{r}} := T^{\mathcal{A}(r)} \setminus R_r}$*. If* $x, y \in T$*, the pair* $(\mathcal{T}, x)$ *is called a* **pointed model***, and* $(\mathcal{T}, x, y)$ *is called a* **two-pointed model***.*

*Note 1.* To simplify the notation when there is no risk of confusion, we will often simply use $r$ to refer to the semantics $R_r$ corresponding to the $\mathcal{R}$-tree currently in discussion.

In some cases when $a$ is an unary symbol, we might simply write $a$ instead of the node expression $\langle\varepsilon\rangle_a$.

*Example 1.* Let us consider $\mathcal{T} = \langle T, E_\downarrow, \{a, b, c, =_n, =_w\}\rangle$ a representation of a very simple and brief bibliographical database as in Fig. 1. Here $a, b, c$ are unary predicates (representing in $\mathcal{T}$ that a node is an author, book, or chapter, respectively. $=_n, =_w$ are binary predicates, which express in $\mathcal{T}$ when two nodes have the same numerical value for $=_n$, or the same word for $=_w$.

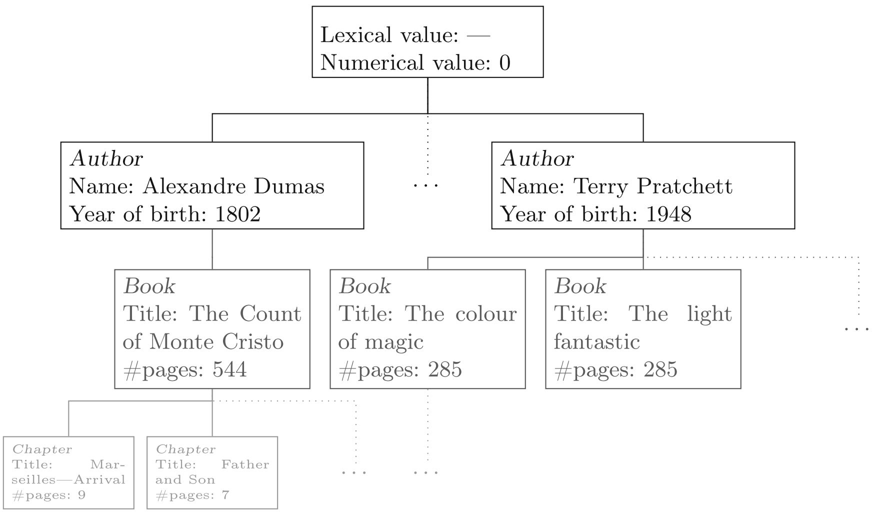


**Fig. 1.** A representation of a bibliographical database as in Example 1, via a tree whose nodes contain both a numeric and lexical value and where nodes can belong to any of three types of unary relations ((a)uthor), (b)ook, (c)hapter).

**Definition 3.** *We now give the interpretation of the symbols from Definition 1 over an $\mathcal{R}$-tree $\mathcal{T} = \langle T, E_\downarrow, \{R_r\}_{r\in\mathcal{R}}\rangle$.*

$$[\![\varepsilon]\!]_\mathcal{T} := \{(x,y) \in T^2 \mid x = y\} \quad [\![\downarrow]\!]_\mathcal{T} := E_\downarrow \quad [\![[\varphi]]\!]_\mathcal{T} := \{(x,x) \in T^2 \mid x \in [\![\varphi]\!]_\mathcal{T}\}$$
$$[\![\alpha\beta]\!]_\mathcal{T} := [\![\alpha]\!]_\mathcal{T} \circ [\![\beta]\!]_\mathcal{T} \quad [\![\alpha \cup \beta]\!]_\mathcal{T} := [\![\alpha]\!]_\mathcal{T} \cup [\![\beta]\!]_\mathcal{T}$$
$$[\![\varphi \wedge \psi]\!]_\mathcal{T} := [\![\varphi]\!]_\mathcal{T} \cap [\![\psi]\!]_\mathcal{T} \quad [\![\neg\varphi]\!]_\mathcal{T} := T \setminus [\![\varphi]\!]_\mathcal{T}$$

$$\begin{aligned}[\![\langle\alpha_1,\alpha_2,\ldots,\alpha_n\rangle_r]\!]_\mathcal{T} := \{x \in T \mid\ & \exists y_1,y_2,\ldots,y_n \in T\ \forall i \in \{1,2,\ldots,n\}\\ & (x,y_i) \in [\![\alpha_i]\!]_\mathcal{T} \wedge R_r(y_1,y_2,\ldots,y_n)\}\\ [\![\langle\alpha_1,\alpha_2,\ldots,\alpha_n\rangle_{\overline{r}}]\!]_\mathcal{T} := \{x \in T \mid\ & \exists y_1,y_2,\ldots,y_n \in T\ \forall i \in \{1,2,\ldots,n\}\\ & (x,y_i) \in [\![\alpha_i]\!]_\mathcal{T} \wedge R_{\overline{r}}(y_1,y_2,\ldots,y_n)\}\end{aligned}$$

If $\varphi$ is a node expression and $\alpha$ a path expression, then we write $(\mathcal{T},x) \models \varphi$ iff $x \in [\![\varphi]\!]_\mathcal{T}$ and $(\mathcal{T},x,y) \models \alpha$ iff $(x,y) \in [\![\alpha]\!]_\mathcal{T}$.

*Remark 1.* Note that we can express the node property $\langle\alpha\rangle$, which indicates that from a node it is possible to descend to some node via the path expression $\alpha$. Indeed, we can see that for any $r \in \mathcal{R}$, the following formula expresses the desired property: $\langle\alpha,\ldots,\alpha\rangle_r \vee \langle\alpha,\ldots,\alpha\rangle_{\overline{r}}$.

We can also define a node expression that is true on any node, irrespective of the semantics of the tree: take any $r \in \mathcal{R}$ and define $\top := \langle\varepsilon,\ldots,\varepsilon\rangle_r \vee \langle\varepsilon,\ldots,\varepsilon\rangle_{\overline{r}}$.

*Example 2.* From the root $t_0$ of the database represented in Fig. 1, we could ask whether there is an author having the same name as some book title. The answer to this depends on whether $(\mathcal{T},t_0) \models \langle\downarrow[\langle\varepsilon\rangle_a], \downarrow\downarrow[\langle\varepsilon\rangle_b]\rangle_{=_w}$. From what we can see in the graphical representation, this does not happen, as we can see that $[\![\langle\downarrow[\langle\varepsilon\rangle_a], \downarrow\downarrow[\langle\varepsilon\rangle_b]\rangle_{=_w}]\!]_\mathcal{T} = \emptyset$.

Note that here (as in some types of data logics) asking whether there are two different nodes with the same data value is in general not possible. For example, $[\![a \wedge \langle\downarrow,\downarrow\rangle_{=_n}]\!]_\mathcal{T}$ (see Note 1) will contain *any* node that corresponds to an author that has written a book, since there is no guarantee that the two witnesses of $\downarrow$ are different. Indeed, in the figure $[\![\langle\varepsilon\rangle_a \wedge \langle\downarrow,\downarrow\rangle_{=_n}]\!]_\mathcal{T}$ would contain the node of Alexandre Dumas, even though it has only one book child.

*Example 3.* Consider an extension of $\mathcal{T}$ and $\mathcal{R}$ including the binary predicate $=^d_n$, such that over $\mathcal{T}$, $x =^d_n y$ iff ($x =_n y$ and $x \neq y$). Now we can express in $\mathcal{T}$ properties such as 'this author has two different books with the same number of pages', via $\varphi : a \wedge \langle\downarrow,\downarrow\rangle_{=^d_n}$.

**Definition 4.** *Two expressions $\eta_1$ and $\eta_2$ of the same type (node or path expression) are said to be* **semantically equivalent**, *written $\boldsymbol{\eta_1 \equiv \eta_2}$ if for all $\mathcal{T}$ we have that $[\![\eta_1]\!]_\mathcal{T} = [\![\eta_2]\!]_\mathcal{T}$.*

*Remark 2.* It easy to see that the semantic equivalence is preserved by any syntactic construction. That is, negating two equivalent node expression results in equivalent node expressions, concatenating a path expression to two different but equivalent path expressions results in two equivalent path expressions, etc.

*Remark 3.* Let $\overline{\gamma} = \gamma_1, \ldots, \gamma_k$ and $\overline{\gamma}' = \gamma_{k+1}, \ldots, \gamma_n$ be finite (and potentially empty) sequences of path expressions. For all $\alpha$ and $\beta$ path expressions and for all $* \in \mathcal{R} \cup \overline{\mathcal{R}}$ with $\mathcal{A}(*) = n+1$ we have that

$$\langle \overline{\gamma}, \alpha \cup \beta, \overline{\gamma}' \rangle_* \equiv \langle \overline{\gamma}, \alpha, \overline{\gamma}' \rangle_* \vee \langle \overline{\gamma}, \beta, \overline{\gamma}' \rangle_*$$

Guided by this fact, we will re-define $\mathbf{XPath}_{\mathcal{R}}(\downarrow)$ as the fragment of the original one (Definition 1) where we do not include the rule $\alpha \cup \beta$ in the grammar. That is, the fragment whose path expressions do not have the symbol $\cup$ in their syntax. Even though this **union-free fragment** is less expressive when considering both path and node expressions, the expressive power remains the same as the full fragment when considering only pointed models and node expressions, because of the semantic equivalence given in Remark 3.

We now give the definition of direct path expressions, which are path expressions without unnecessary concatenations of symbols. This definition is used later to define normal expressions, whose purpose is to simplify the proofs by induction.

**Definition 5.** *A* **direct path expression** $\alpha$ *is a path expression of the form* $\alpha = \varepsilon$ *or* $\alpha = \downarrow \xi_1 \downarrow \ldots \downarrow \xi_n$ *where each* $\xi_i$ *is an empty string or a node test.*

A normal expression is one with all the path expressions in its path tests being direct. We will formally define this idea by means the operator sub$(-)$.

**Definition 6.** *For a formula* $\eta$ *we denote by* ***sub***$(\boldsymbol{\eta})$ *the set defined recursively as follows:*

$$sub(\varepsilon) = sub(\downarrow) := \emptyset \quad sub(\alpha\beta) := sub(\alpha) \cup sub(\beta)$$
$$sub(\neg\varphi) = sub([\varphi]) := sub(\varphi) \quad sub(\varphi \wedge \psi) := sub(\varphi) \cup sub(\psi)$$
$$sub(\langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle_*) := \{\alpha_1, \alpha_2, \ldots, \alpha_n\} \cup \bigcup_{i=1}^{n} sub(\alpha_i) \quad \textit{for } * \in \mathcal{R} \cup \overline{\mathcal{R}}$$

**Definition 7.** *A formula* $\eta$ *is a* **normal expression** *if all the path expressions in* $sub(\eta) \cup \{\eta\}$ *are direct.*

The downward depth of an expression measures the maximum depth from the current point of evaluation that the formula could potentially 'see'. The idea is that, when analysing such an expression over a particular point or pair of points, nodes that are further down than this depth have no effect on the resulting truth value of the expression.

**Definition 8.** *The* **downward depth** *of* $\eta$ *denoted by* ***dd***$(\boldsymbol{\eta})$ *is the number defined as follows:*

$$dd(\neg\varphi) := dd(\varphi) \quad dd(\varphi \wedge \psi) := \max\{dd(\varphi), dd(\psi)\}$$
$$dd(\lambda) := 0 \quad \textit{where } \lambda \textit{ represents the empty string}$$
$$dd(\varepsilon\beta) := dd(\beta) \quad dd([\varphi]\beta) := \max\{dd(\varphi), dd(\beta)\} \quad dd(\downarrow \beta) := 1 + dd(\beta)$$
$$dd(\langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle_*) := \max\{dd(\alpha_1), dd(\alpha_2), \ldots, dd(\alpha_n)\} \quad \textit{for } * \in \mathcal{R} \cup \overline{\mathcal{R}}$$

*The set of all formulas with downward depth less than or equal to $\ell \geq 0$ is written as* $\mathbf{XPath}_{\mathcal{R}}(\downarrow)$.

Note that this definition encompasses all (union-free) formulas in XPath$_{\mathcal{R}}(\downarrow)$ and the function dd$(-)$ is well-defined.

*Remark 4.* Let $\overline{\gamma} = \gamma_1, \ldots, \gamma_k$ and $\overline{\gamma}' = \gamma_{k+1}, \ldots, \gamma_n$ be finite (and potentially empty) sequences of path expressions. The following semantic equivalences also preserve the downward depth.

1. $\langle \overline{\gamma}, [\varphi]\alpha, \overline{\gamma}' \rangle_* \equiv \varphi \wedge \langle \overline{\gamma}, \alpha, \overline{\gamma}' \rangle_*$ $\qquad * \in \mathcal{R} \cup \overline{\mathcal{R}}$ with $\mathcal{A}(*) = n + 1$
2. $\varepsilon\alpha \equiv \alpha$
3. $[\varphi][\psi] \equiv [\varphi \wedge \psi]$

**Proposition 1.** *For every formula $\eta$ we have that*

*i) If $\eta$ is a node expression then there exists a normal node expression $\eta'$ with $dd(\eta) = dd(\eta')$ and $\eta \equiv \eta'$.*
*ii) If $\eta$ is a path expression then there exists a normal path expression $\eta'$ with $dd(\eta) = dd(\eta')$ and $\eta \equiv \eta'$.*

*Proof.* One can easily prove the statement by syntactic induction over $\eta$ and making use of the semantic equivalences from Remark 4.

## 3 Bisimulation and Equivalence

The classic Hennessy-Milner's characterization theorem [6] for Basic Modal Logic establishes the relation between two notions: logical equivalence and bisimilarity. In our case, the former notion indicates when a pair of pointed models are indistinguishable by means of node expressions. The latter intuitively ensures that for each selection of paths in one of the models, there are copies in the other, preserving the possible relational properties between their respective final points.

**Definition 9.** *Let $\ell \geq 0$. Given $(\mathcal{T}, x)$ and $(\mathcal{T}', x')$ we say that they are* $\boldsymbol{\mathcal{R}_\ell}$**-logically equivalent** *and denote it by* $\boldsymbol{(\mathcal{T}, x) \equiv_\ell^{\mathcal{R}} (\mathcal{T}', x')}$ *if for all node expression $\varphi \in XPath_{\mathcal{R}}^{\ell}(\downarrow)$ we have that $(\mathcal{T}, x) \models \varphi$, if and only if, $(\mathcal{T}', x') \models \varphi$.*

*$(\mathcal{T}, x)$ and $(\mathcal{T}', x')$ are* $\boldsymbol{\mathcal{R}}$**-logically equivalent**, *written* $\boldsymbol{(\mathcal{T}, x) \equiv^{\mathcal{R}} (\mathcal{T}', x')}$, *if* $\boldsymbol{(\mathcal{T}, x) \equiv_\ell^{\mathcal{R}} (\mathcal{T}', x')}$ *for all $\ell \geq 0$. In other words, if for any node expression $\varphi$, $(\mathcal{T}, x) \models \varphi$, if and only if, $(\mathcal{T}', x') \models \varphi$.*

**Definition 10.** *A* **path** *$\mu$ in a tree $\mathcal{T}$ is a sequence $\mu = \mu_0 \downarrow \mu_1 \downarrow \ldots \downarrow \mu_n$ where $n \geq 0$, $\mu_i \in T$ for all $i \in \{0, \ldots, n\}$ and $E_\downarrow(\mu_i, \mu_{i+1})$ for all $i \in \{0, 1, \ldots, n-1\}$. The length* $\boldsymbol{len(\mu) := n}$ *is the number of symbols $\downarrow$ in $\mu$. The i-th node of $\mu$ is* $\boldsymbol{[\mu]_i := \mu_i}$ *and* $\boldsymbol{end(\mu) := [\mu]_{len}(\mu)}$ *is the final node of $\mu$.*

*We denote by* $\boldsymbol{Path(\mathcal{T})}$ *the set of all paths $\mu$ in $\mathcal{T}$. For a node $x \in \mathcal{T}$,* $\boldsymbol{Path(\mathcal{T}, x)}$ *are the paths $\mu$ starting from the node $x$, i.e., $[\mu]_0 = x$ and by* $\boldsymbol{Path_k(\mathcal{T}, x)}$ *we refer to the subset of paths in $Path(\mathcal{T}, x)$ with length at most $k$.*

*The concatenation* $\boldsymbol{\mu} \odot \boldsymbol{\nu}$ *of two paths* $\mu, \nu \in Path(\mathcal{T})$ *such that* $\nu_0 = end(\mu)$ *is defined as* $[\mu \odot \nu]_i := [\mu]_i$ *for all* $i \in \{0, \ldots, len(\mu)\}$ *and* $[\mu \odot \nu]_{len(\mu)+i} := [\nu]_i$ *for all* $i \in \{0, \ldots, len(\nu)\}$*.*

**Definition 11.** *Given a* $\mathcal{R}$*-tree* $\mathcal{T}$ *and a node* $x \in T$*,* $\boldsymbol{\mathcal{T}|^x_\ell}$ *is the* $\mathcal{R}$*-tree whose underlying tree is the set of nodes* $y \in T$ *for which there exists a path* $\mu \in Path_\ell(\mathcal{T}, x)$ *with* $end(\mu) = y$ *(note that* $x$ *is the root of such tree).*

*Remark 5.* For all $\ell \geq 0$ and $\mathcal{R}$-tree $\mathcal{T}$, we have that $(\mathcal{T}|^x_\ell, x) \equiv^{\mathcal{R}}_\ell (\mathcal{T}, x)$.

**Definition 12.** *Let* $\mathcal{T}$ *and* $\mathcal{T}'$ *be* $\mathcal{R}$*-trees. An* $\boldsymbol{\mathcal{R}_\ell}$**-bisimulation** $\mathcal{Z} = \{Z_k\}_{0 \leq k \leq \ell}$ *is a family of relations* $Z_k \subseteq T \times T'$ *such that for all* $k$*, for all* $(x, x') \in Z_k$*, and for all* $n \in Im(\mathcal{A})$*, the clauses below hold.*

**Zig** *For every selection of paths* $\mu_1, \mu_2, \ldots, \mu_n \in Path_k(\mathcal{T}, x)$*, there exist paths* $\mu'_1, \mu'_2, \ldots, \mu'_n \in Path_k(\mathcal{T}', x')$ *such that for all* $j \in \{1, \ldots, n\}$ *and for all* $r \in \mathcal{R}$ *we have that:*
- *i)* $len(\mu_j) = len(\mu'_j)$
- *ii)* $([\mu_j]_i, [\mu'_j]_i) \in Z_{k-i}\ \forall i \in \{0, \ldots, len(\mu_j)\}$
- *iii)* $R_r(end(\mu_1), end(\mu_2), \ldots, end(\mu_n)) \Leftrightarrow R'_r(end(\mu'_1), end(\mu'_2), \ldots, end(\mu'_n))$

**Zag** *For every selection of paths* $\mu'_1, \mu'_2, \ldots, \mu'_n \in Path_k(\mathcal{T}', x')$ *there exist paths* $\mu_1, \mu_2, \ldots, \mu_n \in Path_k(\mathcal{T}, x)$ *such that for all* $j \in \{1, \ldots, n\}$ *and for all* $r \in \mathcal{R}$ *we have that:*
- *i)* $len(\mu_j) = len(\mu'_j)$
- *ii)* $([\mu_j]_i, [\mu'_j]_i) \in Z_{k-i}\ \forall i \in \{0, \ldots, len(\mu_j)\}$
- *iii)* $R_r(end(\mu_1), end(\mu_2), \ldots, end(\mu_n)) \Leftrightarrow R'_r(end(\mu'_1), end(\mu'_2), \ldots, end(\mu'_n))$

*Two pointed models* $(\mathcal{T}, x)$ *and* $(\mathcal{T}', x')$ *are said to be* $\boldsymbol{\mathcal{R}_\ell}$**-bisimilar***, denoted* $\boldsymbol{(\mathcal{T}, x) \leftrightharpoons^{\mathcal{R}}_\ell (\mathcal{T}', x')}$*, if there exists an* $\mathcal{R}_\ell$*-bisimulation* $\mathcal{Z} = \{Z_i\}_{0 \leq i \leq \ell}$ *such that* $(x, x') \in Z_\ell$*.*

*If* $Z \subseteq T \times T'$ *is a relation such that for all* $\ell \geq 0$ *the family* $\{Z_i \mid Z_i = Z\}_{0 \leq i \leq \ell}$ *is a* $\mathcal{R}_\ell$*-bisimulation then we call* $Z$ *an* $\boldsymbol{\mathcal{R}}$**-bisimulation***. If there exists a* $\mathcal{R}$*-bisimulation* $Z$ *with* $(x, x') \in Z$ *then* $(\mathcal{T}, x)$ *and* $(\mathcal{T}', x')$ *are* $\boldsymbol{\mathcal{R}}$**-bisimilar***, written as* $\boldsymbol{(\mathcal{T}, x) \leftrightharpoons^{\mathcal{R}} (\mathcal{T}', x')}$*.*

*Remark 6.* It is useful to observe that, when there are predicates of arity 2 or greater, the Zig (and Zag) conditions can be replaced in the case of unary predicates with a simpler 'Harmony' condition in the style of bisimulation for modal logics and XPath$_=$: for any unary predicate $u$ it is enough to check that whenever $xZx'$, then $u(x)$ iff $u(x')$. Note, however, that this replacement cannot be done if we *only* have unary predicates in $\mathcal{R}$, since doing so would remove all Zig and Zag conditions, and thus we would not be comparing any topological information about the models.

*Remark 7.* Using the Remark 6, we can see that our Definition 12 for the concept of $\mathcal{R}$-bisimulation generalizes the definition of bisimulation for XPath$_=(\downarrow)$ over the universe of data trees from [4]. It does so by taking $\mathcal{R} = \{=_d\} \cup \mathbb{A}$, where $=_d$ is a binary symbol (interpreted as data equality over data trees) and the finite symbols in the label set $\mathbb{A}$ are unary predicates. See also Theorem 4 and the discussion preceding it.

*Remark 8.* The notion of $\mathcal{R}$-bisimilarity given in Definition 12 does not coincide with bisimilarity for multi-relational Kripke models (where $\downarrow$ and each relation from $\mathcal{R}$ get their own modal operator, and where we translate from $\mathcal{R}$-trees into Kripke models). Indeed, consider the case where $\mathcal{R}$ consists solely of a binary relation $=_d$ (we will represent the semantics of $=_d$ with numeric data values). For an example of two pointed models that are modally bisimilar but not $\mathcal{R}$-bisimilar, consider: on one hand the infinite linear tree $(\mathcal{T}, x)$, where the root $x$ has data value 1, the sole next child has data value 2, the next one has data value 1, and so on alternating between these two values (i.e. $1, 2, 1, 2, \ldots$); on the other hand, take the infinite linear tree $(\mathcal{T}', x')$ that has data value 1 in all nodes (i.e. $1, 1, 1, 1, \ldots$).

Intuitively, while modal bisimulation appears to have greater navigational freedom by being able to move with any modality from $\mathcal{R}$, when doing that it cannot keep track of the actual topology of the model (given by $\downarrow$) nor can it ask whether the endpoints of paths are related via an $r \in \mathcal{R}$.

*Example 4.* Let $\mathcal{R} = \{b, f, S\}$, where $b, f$ are unary predicate symbols and $S$ is a ternary predicate symbol. We consider the $\mathcal{R}$-trees $\mathcal{T}, \mathcal{T}'$ from Fig. 2, where a node in $b$ is represented as having a red border, and a node satisfying $f$ is represented as being filled with the color black. In both trees, $S$ has an interpretation related to the numbers: $S(x, y, z)$ iff $d(x) = d(y) + d(z)$, where $d(w)$ represents the number drawn on the node $w$. If we call $t_0, t'_0$ the respective roots of both trees, it is easy to verify that $(\mathcal{T}, t_0)$ and $(\mathcal{T}', t'_0)$ are $\mathcal{R}$-bisimilar via the represented $Z$ relation. It is important to note that it does not matter that $\mathcal{T}$ represents the values of nodes in integers and $\mathcal{T}'$ does so with rational numbers: the only thing that matters is the semantics of each predicate in $\mathcal{R}$, as the logic itself has no way of 'seeing' these particular values (and indeed some possible semantics of $S$ cannot be expressed in this form).

As we have for modal logics, the bounded notions $\mathcal{R}_\ell$-logical equivalence and $\mathcal{R}_\ell$-bisimulation coincide. That is, two pointed models are $\mathcal{R}_\ell$-logically equivalent, if and only if, they are $\mathcal{R}_\ell$-bisimilar.

The proof of each one of the following statements can be found in the Appendix.

**Lemma 1.** *For every $\ell \geq 0$, there are finitely many equivalence classes (modulo semantic equivalence) of node expressions with downward depth at most $\ell$.*

**Corollary 1.** *For each $\ell \geq 0$ and a pointed model $(\mathcal{T}, x)$ there is a node expression $\chi^\ell_{(\mathcal{T},x)} \in \mathit{XPath}^\ell_{\mathcal{R}}(\downarrow)$ satisfying that for any pointed model $(\mathcal{T}', x')$, $(\mathcal{T}, x) \equiv^{\mathcal{R}}_\ell (\mathcal{T}', x')$ iff $(\mathcal{T}', x') \models \chi^\ell_{(\mathcal{T},x)}$.*

**Proposition 2.** *Given $(\mathcal{T}, x)$ and $(\mathcal{T}', x')$ we have that if $(\mathcal{T}, x) \leftrightarroweq^{\mathcal{R}}_\ell (\mathcal{T}', x')$ then $(\mathcal{T}, x) \equiv^{\mathcal{R}}_\ell (\mathcal{T}', x')$.*

**Proposition 3.** *Given $(\mathcal{T}, x)$ and $(\mathcal{T}', x')$ we have that if $(\mathcal{T}, x) \equiv^{\mathcal{R}}_\ell (\mathcal{T}', x')$ then $(\mathcal{T}, x) \leftrightarroweq^{\mathcal{R}}_\ell (\mathcal{T}', x')$.*

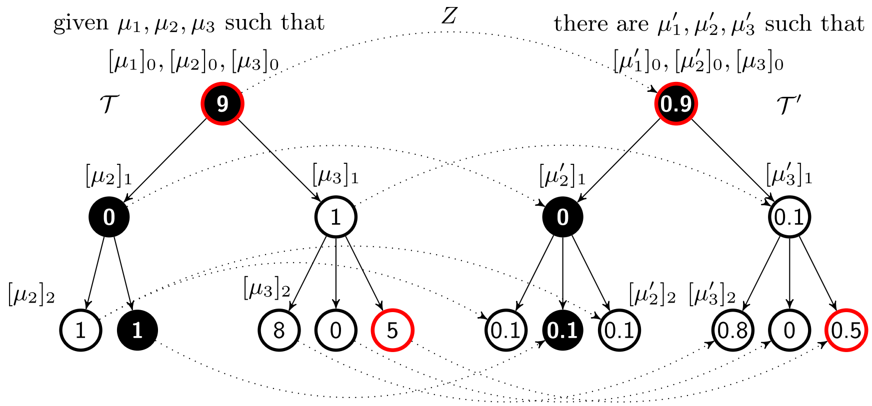


**Fig. 2.** A representation of two $\mathcal{R}$-trees $\mathcal{T}$ and $\mathcal{T}'$ for $\mathcal{R} = \{b, f, S\}$, as in Example 4. Also represented in the figure is a Zig step and, via a dotted line connecting nodes, a $\{b, f, S\}$-bisimulation $Z$ between $(\mathcal{T}, t_0)$ and $(\mathcal{T}', t'_0)$.

We can unify Proposition 2 and Proposition 3 in the following statement:

**Theorem 1.** *Given* $(\mathcal{T}, x)$ *and* $(\mathcal{T}', x')$ *we have that* $(\mathcal{T}, x) \leftrightharpoons_\ell^{\mathcal{R}} (\mathcal{T}', x')$, *if and only if,* $(\mathcal{T}, x) \equiv_\ell^{\mathcal{R}} (\mathcal{T}', x')$.

## 4 Characterizations

In this section we state and prove characterization results for $\text{XPath}_{\mathcal{R}}(\downarrow)$, in the style of Hennessy-Milner's theorem [6] for Basic Modal Logic. Our result states that $\mathcal{R}$-logical equivalence and $\mathcal{R}$-bisimulation agree over models whose underlying tree is finitely branching. After giving this result, we will treat $\text{XPath}_{\mathcal{R}}(\downarrow)$ as a fragment of the first order logic in order to show a characterization result in the style of van Benthem's theorem [7] for Basic Modal Logic, concluding that the formulas of $\text{XPath}_{\mathcal{R}}(\downarrow)$ are exactly those of the first order logic (with certain signature) which are preserved by $\mathcal{R}$-bisimulation.

**Proposition 4.** *Given* $(\mathcal{T}, x)$ *and* $(\mathcal{T}', x')$ *we have that if* $(\mathcal{T}, x) \leftrightharpoons^{\mathcal{R}} (\mathcal{T}', x')$ *then* $(\mathcal{T}, x) \equiv^{\mathcal{R}} (\mathcal{T}', x')$.

*Proof.* Suppose $(\mathcal{T}, x) \leftrightharpoons^{\mathcal{R}} (\mathcal{T}', x')$ via $(x, x') \in Z \subseteq T \times T'$. For every $\ell \geq 0$, the family $\mathcal{Z}_\ell := \{Z_0 = Z_1 = \cdots = Z_\ell := Z\}$ is a $\mathcal{R}_\ell$-bisimulation between $(\mathcal{T}, x)$ and $(\mathcal{T}', x')$. Thus, by Theorem 1, for every $\ell \geq 0$ we have that $(\mathcal{T}, x) \equiv_\ell^{\mathcal{R}}$. In other words, $(\mathcal{T}, x) \equiv^{\mathcal{R}} (\mathcal{T}', x')$.

**Definition 13.** *A pointed model* $(\mathcal{T}, x)$ *is* **finitely branching** *if for all* $k \geq 0$ *the set* $Path_k(\mathcal{T}, x)$ *is finite.*

**Theorem 2 (Hennessy-Milner's style characterization).** *If* $(\mathcal{T}, x)$ *and* $(\mathcal{T}, x')$ *are finitely branching, then* $(\mathcal{T}, x) \leftrightharpoons^{\mathcal{R}} (\mathcal{T}', x')$ *if and only if* $(\mathcal{T}, x) \equiv^{\mathcal{R}} (\mathcal{T}', x')$.

*Proof.*

(⇒) By Proposition 4.
(⇐) Consider the relation $Z \subseteq T \times T'$, which is defined such that $(z, z') \in Z$ iff $(\mathcal{T}, z) \equiv^{\mathcal{R}} (\mathcal{T}', z')$. The proof to show that $Z$ is a $\mathcal{R}$-bisimulation is analogous to that which was given for Proposition 3 in order to see that $\mathcal{Z}$ was a $\mathcal{R}_\ell$-bisimulation. Now, we define $\Phi^{(j,i)} := \bigwedge_{\overline{\mu} \in P} \varphi_{\overline{\mu}}^{(j,i)}$ since the set $P$ is finite because $(\mathcal{T}', x')$ is finitely branching.

*Remark 9.* The classical counterexample for the modal logic with only one modality still works to see that the finitely branching hypothesis is required: take on one side an infinitely branching tree constructed with one branch of each finite length; on the other side, a copy of the previous tree with the addition of an infinitely long linear branch hanging from the root. For every $r \in \mathcal{R}$, we set in both trees that $R_r = \emptyset$. It can be seen that both $\mathcal{R}$-trees with their roots are logically equivalent but that any proposed $Z$ fails to be a bisimulation.

From now on, we focus on XPath$_{\mathcal{R}}(\downarrow)$ as a fragment of the first-order logic FO$(\sigma_{\mathcal{R}})$ with the natural signature $\sigma_{\mathcal{R}}$, and with a standard translation ST$(-)$ between formulas of XPath$_{\mathcal{R}}(\downarrow)$ and of first-order logic. For details, see the corresponding Definitions 16 and 17 in the Appendix.

The following lemmas are the key for proving the van Benthem's characterization style theorem. Before stating them, we need to give some definitions:

**Definition 14.** *Let $\ell \geq 0$ and $\psi(u), \varphi(u) \in FO(\sigma_{\mathcal{R}})$ with one free variable $u$.*

- *$\psi(u)$ is* **$\ell$-local** *if for all pointed model $(\mathcal{T}, x)$, $\mathcal{T} \models \psi[u \mapsto x]$ iff $\mathcal{T}|_\ell^x \models \psi[u \mapsto x]$.*
- *$\psi(u)$ is* **$\leftrightharpoons^{\mathcal{R}}$-invariant** *if for all $(\mathcal{T}, x)$ and $(\mathcal{T}', x')$ such that $\mathcal{T} \models \psi[u \mapsto x]$ and $(\mathcal{T}, x) \leftrightharpoons^{\mathcal{R}} (\mathcal{T}', x')$ then $\mathcal{T}' \models \psi[u \mapsto x']$.*
- *$\psi(u)$ is* **$\leftrightharpoons_\ell^{\mathcal{R}}$-invariant** *if for all $(\mathcal{T}, x)$ and $(\mathcal{T}', x')$ such that $\mathcal{T} \models \psi[u \mapsto x]$ and $(\mathcal{T}, x) \leftrightharpoons_\ell^{\mathcal{R}} (\mathcal{T}', x')$ then $\mathcal{T}' \models \psi[u \mapsto x']$.*
- *$\psi(u)$ and $\varphi(u)$ are* **semantically equivalent over trees $\psi(u) \overset{trees}{\equiv} \varphi(u)$** *if for all $(\mathcal{T}, x)$ we have that $\mathcal{T} \models \psi[u \mapsto x]$ iff $\mathcal{T} \models \varphi[u \mapsto x]$.*
- *A pointed model $(\mathcal{T}, x)$ has* **depth at most $\ell$** *if $Path_\ell(\mathcal{T}, x) = Path_n(\mathcal{T}, x)$ for all $n \geq \ell$.*

**Lemma 2.** *For each $\leftrightharpoons^{\mathcal{R}}$-invariant $\psi(u) \in FO(\sigma_{\mathcal{R}})$ with one free variable $u$, there is $\ell \geq 0$ such that $\psi$ is $\ell$-local.*

*Proof.* See Appendix.

**Lemma 3.** *Suppose $(\mathcal{T}, x)$ and $(\mathcal{T}', x')$ have depth at most $\ell$. Then $(\mathcal{T}, x) \leftrightharpoons^{\mathcal{R}} (\mathcal{T}', x')$ if and only if $(\mathcal{T}, x) \leftrightharpoons_\ell^{\mathcal{R}} (\mathcal{T}', x')$. Equivalently, $(\mathcal{T}, x) \leftrightharpoons^{\mathcal{R}} (\mathcal{T}', x')$, if and only if, $(\mathcal{T}, x) \equiv_\ell^{\mathcal{R}} (\mathcal{T}', x')$.*

*Proof.* The left-to-right direction is clear. For the other direction, suppose that we have $(\mathcal{T}, x) \leftrightharpoons_\ell^{\mathcal{R}} (\mathcal{T}', x')$ via $\mathcal{Z} = \{Z_k\}_{0 \leq k \leq \ell}$. Then we have that $Z := \bigcup_{k=0}^{\ell} Z_k$ is a $\mathcal{R}$-bisimulation between $(\mathcal{T}, x)$ and $(\mathcal{T}', x')$.

**Theorem 3 (van Benthem's style characterization).** *Let $\psi(u)$ be a formula in $FO(\sigma_{\mathcal{R}})$ with one free variable $u$. The following statements are equivalent:*

1. *$\psi(u)$ is $\leftrightharpoons^{\mathcal{R}}$-invariant.*
2. *There exists a node expression $\varphi$ such that $\psi(u) \overset{trees}{\equiv} ST(\varphi)(u)$*

*Proof.*

(1⇒2) Suppose $\psi(u)$ is $\leftrightharpoons^{\mathcal{R}}$-invariant. By Lemma 2 we know that there is some $\ell \geq 0$ for which $\psi(u)$ is $\ell$-local. Let us see that $\psi(u)$ is $\leftrightharpoons^{\mathcal{R}}_{\ell}$-invariant for such $\ell$: given $(\mathcal{T}, x)$ and $(\mathcal{T}', x')$ such that $\mathcal{T} \models \psi[u \mapsto x]$ and $(\mathcal{T}, x) \leftrightharpoons^{\mathcal{R}}_{\ell} (\mathcal{T}', x')$ we want to show that $\mathcal{T}' \models \psi[u \mapsto x']$. On one side, by Remark 5, $(\mathcal{T}|^{x}_{\ell}, x) \leftrightharpoons^{\mathcal{R}}_{\ell} (\mathcal{T}, x) \leftrightharpoons^{\mathcal{R}}_{\ell} (\mathcal{T}', x') \leftrightharpoons^{\mathcal{R}}_{\ell} (\mathcal{T}'|^{x'}_{\ell}, x')$. On the other side, by Lemma 3, $(\mathcal{T}|^{x}_{\ell}, x) \equiv^{\mathcal{R}} (\mathcal{T}'|^{x'}_{\ell}, x')$. Since $\mathcal{T} \models \psi[u \mapsto x]$ iff $\mathcal{T}|^{x}_{\ell} \models \psi[u \mapsto x]$, we conclude that $\mathcal{T}'|^{x'}_{\ell} \models \psi[u \mapsto x']$ iff $\mathcal{T}' \models \psi[u \mapsto x']$. From this, it is clear that $\varphi(u) \overset{trees}{\equiv} \mathrm{ST}\left(\bigvee_{\mathcal{T} \models \varphi[u \mapsto x]} \chi^{\ell}_{(\mathcal{T}, x)}\right)(u)$ where $\chi^{\ell}_{(\mathcal{T}, x)}$ is given by Corollary 1.

(2⇒1) Suppose $\psi(u) \overset{trees}{\equiv} \mathrm{ST}(\varphi)(u)$. Using Proposition 4, we can see that $\mathrm{ST}(\varphi)(u)$ is $\leftrightharpoons^{\mathcal{R}}$-invariant. Indeed, if $(\mathcal{T}, x) \leftrightharpoons^{\mathcal{R}} (\mathcal{T}', x')$, then, by Proposition 4, $(\mathcal{T}, x) \equiv^{\mathcal{R}} (\mathcal{T}', x')$; also if $\mathcal{T} \models \mathrm{ST}(\varphi)[u \mapsto x]$, because $\mathrm{ST}(-)$ is truth-preserving then $(\mathcal{T}, x) \models \varphi$. Hence, since $(\mathcal{T}, x) \equiv^{\mathcal{R}} (\mathcal{T}', x')$, we have that $(\mathcal{T}', x') \models \varphi$ which is equivalent to $\mathcal{T}' \models \mathrm{ST}(\varphi)[u \mapsto x']$. So $\psi(u)$ is indeed $\leftrightharpoons^{\mathcal{R}}$-invariant.

So far, we have considered that $\mathcal{R}$ had no fixed semantics on the universe, but for many cases it is reasonable to consider some restrictions, such that a particular binary relation is an equivalence relation, which is a reasonable assumption if we want to talk about $\mathcal{R}$-trees as a generalization of data trees.

Let $\mathbb{A}$ be a subset of unary symbols from $\mathcal{R}$, and let $X$, $S$, and $T$ be three subsets of binary symbols from $\mathcal{R}$. We denote by $\boldsymbol{\mathcal{U}^{\mathcal{R}}_{\mathbb{A},X,S,T}}$ the class of $\mathcal{R}$-trees $\mathcal{T}$ satisfying that for every node $x \in T$ there is a unique symbol $a \in \mathbb{A}$ such that $x \in R_a$, and for all $r \in \mathcal{R}$ the relation $R_r$ is reflexive if $r \in X$, symmetric if $r \in S$ and transitive if $r \in T$. It can be seen that each formula $\psi(u)$ in $\mathrm{FO}(\sigma_{\mathcal{R}})$ that is invariant by bisimulation between structures in $\mathcal{U}^{\mathcal{R}}_{\mathbb{A},X,S,T}$ will be semantically equivalent (relative to $\mathcal{U}^{\mathcal{R}}_{\mathbb{A},X,S,T}$) to the translation of a node expression in $\mathrm{XPath}_{\mathcal{R}}(\downarrow)$. The main reason is that the $\mathcal{R}$-trees $\mathcal{T}_A$ and $\mathcal{T}_B$ constructed in the proof of Lemma 2 are now in the class $\mathcal{U}^{\mathcal{R}}_{\mathbb{A},X,S,T}$, since each binary $\tilde{R}_r$ inherits good properties from $R_r$ as reflexivity, symmetry and transitivity (for the fresh nodes $t_A$ and $t_B$, we can freely assign them to any single unary set $R_a$ with $a \in \mathbb{A}$). The invariance of $\psi(u)$ between structures in this new class allows us to finish the proof. Thus, we can relativize van Benthem's characterization to $\mathcal{U}^{\mathcal{R}}_{\mathbb{A},X,S,T}$. This gives us a generalization to the van Benthem's characterization for $\mathrm{XPath}_{=}(\downarrow)$, where the symbols in $\mathbb{A}$ are the labels and there is a symbol $=_d \in X \cap S \cap T$ whose semantic is the pair of nodes with same data value. Therefore, we have:

**Theorem 4.** *Let $\psi(u)$ be a formula in $FO(\sigma_{\mathcal{R}})$ with one free variable $u$. The following statements are equivalent:*

1. *For all $(\mathcal{T}, x)$ and $(\mathcal{T}', x')$ with $\mathcal{T}, \mathcal{T}' \in \mathcal{U}^{\mathcal{R}}_{\mathbb{A},X,S,T}$, if $\mathcal{T} \models \psi[u \mapsto x]$ and $(\mathcal{T}, x) \leftrightarrows^{\mathcal{R}} (\mathcal{T}', x')$ then $\mathcal{T}' \models \psi[u \mapsto x']$*
2. *There exists a node expression $\varphi \in XPath_{\mathcal{R}}(\downarrow)$ such that for all $(\mathcal{T}, x)$ with $\mathcal{T} \in \mathcal{U}^{\mathcal{R}}_{\mathbb{A},X,S,T}$ it happens that $(\mathcal{T}, x) \models \psi(u)$ iff $(\mathcal{T}, x) \models ST(\varphi)(u)$.*

## 5 Conclusions and Future Work

We have provided (Definition 12) $\mathcal{R}$-bisimulation notions for generalizations of the logic $\text{XPath}_{=}(\downarrow)$ over the full universe of $\mathcal{R}$-trees. This notion coincides with that of [4] for an adequate $\mathcal{R}$ (Remark 7). We have shown that $\mathcal{R}$-bisimulation coincides with $\mathcal{R}$-logical equivalence over finitely branching pointed trees (Theorem 2). We have also characterized $\text{XPath}_{\mathcal{R}}(\downarrow)$ as the fragment of first-order logic that is invariant by $\mathcal{R}$-bisimulations (Theorem 3). Finally, we have shown (Theorem 4) that our results also apply for universes of trees with some restricted semantics which contain the case of $\text{XPath}_{=}(\downarrow)$ over data trees.

In the future, we would like to further generalize the work done in this paper, for example allowing other navigational modalities (such as sibling operators) for trees.

**Acknowledgments.** We thank Román Sasyk (Departamento de Matemática, Universidad de Buenos Aires) for his helpful questions and observations which spurred the main topic of this work.

## A Proofs and Definitions Omitted from the Main Text

**Lemma 1.** For every $\ell \geq 0$, there are finitely many equivalence classes (modulo semantic equivalence) of node expressions with downward depth at most $\ell$.

*Proof.* Without loss of generality (by Proposition 1), we assume that every node expression $\varphi$ is normal, and even more, of the form $\varphi = \langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle_*$ for $* \in \mathcal{R} \cup \overline{\mathcal{R}}$, because every normal expression is a Boolean combination of these. Therefore, if the number of equivalence classes is finite for this type of node expressions, it will also be finite for their Boolean combinations.

The proof goes by induction over $\ell \geq 0$.

$\ell = 0$ For this case, as $\text{dd}(\varphi) = 0$ and $\varphi$ is normal, necessarily $\varphi = \langle \varepsilon, \varepsilon, \ldots, \varepsilon \rangle_*$ with $* \in \mathcal{R} \cup \overline{\mathcal{R}}$ (notice that we can not have path tests like, for instance, $\langle [\langle \varepsilon \rangle], \ldots, [\langle \varepsilon \rangle] \rangle_*$ because $[\langle \varepsilon \rangle]$ is not direct). Since $\mathcal{R}$ is finite, then there are finitely many $\varphi$.

$\ell > 0$ $\varphi = \langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle_*$ for $* \in \mathcal{R} \cup \overline{\mathcal{R}}$ and $0 < \text{dd}(\varphi) \leq \ell$. As the $\alpha_i$ are direct (since $\varphi$ is normal), each node test in one of the $\alpha_i$ has downward depth at most $\ell - 1$. By induction hypothesis, there are finitely many direct $\alpha_i$ modulo semantic equivalence, and, since $\mathcal{R}$ is finite, also there are finitely many $\varphi$ modulo semantic equivalence.

**Corollary 1.** For each $\ell \geq 0$ and a pointed model $(\mathcal{T}, x)$ there is a node expression $\chi^{\ell}_{(\mathcal{T},x)} \in \text{XPath}^{\ell}_{\mathcal{R}}(\downarrow)$ satisfying that for any pointed model $(\mathcal{T}', x')$, $(\mathcal{T}, x) \equiv^{\mathcal{R}}_{\ell} (\mathcal{T}', x')$ iff $(\mathcal{T}', x') \models \chi^{\ell}_{(\mathcal{T},x)}$.

*Proof.* Let $\chi^{\ell}_{(\mathcal{T},x)}$ be the conjunction of all the node expressions modulo equivalence $\varphi \in \text{XPath}^{\ell}_{\mathcal{R}}(\downarrow)$ that $(\mathcal{T}, x)$ satisfies. Lemma 1 ensures that $\chi^{\ell}_{(\mathcal{T},x)}$ is well-defined because it is a conjunction of finitely many node expressions (modulo semantic equivalence), and clearly, it satisfies the desired property.

*Remark 10.* One can redefine some satisfiability notions with the Definition 15:

1. $(\mathcal{T}, x, y) \models \alpha$, if and only if, there exists $\mu \in \text{Path}(\mathcal{T}, x)$ with $\text{end}(\mu) = y$ and $(\mathcal{T}, \mu) \models \alpha$.
2. If $* \in \mathcal{R} \cup \overline{\mathcal{R}}$ then $(\mathcal{T}, x) \models \langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle_*$, if and only if, for all $j \in \{1, 2, \ldots, n\}$ there are paths $\mu_j \in \text{Path}(\mathcal{T}, x)$ such that $(\mathcal{T}, \mu_j) \models \alpha_j$ and also $R_*(\text{end}(\mu_1), \ldots, \text{end}(\mu_n))$

The following definition is a more concrete form of the satisfiability notion for path expressions in an $\mathcal{R}$-tree. It will be used in Lemma 4 and simplify its proof.

**Definition 15.** *Given a path $\mu \in Path(\mathcal{T}, x)$, we define inductively the meaning of $(\mathcal{T}, \mu) \models \gamma$ for a path expression $\gamma$.*

$$\begin{aligned}
(\mathcal{T}, \mu) \models \varepsilon &\overset{def}{\iff} len(\mu) = 0 \\
(\mathcal{T}, \mu) \models [\varphi] &\overset{def}{\iff} \mu = x \text{ and } (\mathcal{T}, x) \models \varphi \\
(\mathcal{T}, \mu) \models \downarrow &\overset{def}{\iff} len(\mu) = 1 \\
(\mathcal{T}, \mu) \models \alpha\beta &\overset{def}{\iff} \text{there is a decomposition } \mu = \mu_\alpha \odot \mu_\beta \\
&\qquad\quad \text{such that } (\mathcal{T}, \mu_\alpha) \models \alpha \text{ and } (\mathcal{T}, \mu_\beta) \models \beta
\end{aligned}$$

**Lemma 4.** *Let $\mathcal{Z} = \{Z_i\}_{0 \leq i \leq \ell}$ be a $\mathcal{R}_\ell$-bisimulation between $\mathcal{T}$ and $\mathcal{T}'$. For all $k \in \{0, 1, \ldots, \ell\}$, any normal expression $\eta$ with $dd(\eta) \leq k$, and paths $\mu \in Path_k(\mathcal{T}, x)$ and $\mu' \in Path_k(\mathcal{T}', x')$ such that $([\mu]_i, [\mu']_i) \in Z_{k-i}$ for all $i \in \{0, 1, \ldots, k\}$, we have that:*

1. *If $\eta$ is a node expression then $(\mathcal{T}, x) \models \eta$, if and only if, $(\mathcal{T}', x') \models \eta$.*
2. *If $\eta$ is a path expression then $(\mathcal{T}, \mu) \models \eta$, if and only if, $(\mathcal{T}', \mu') \models \eta$.*

*Proof.* The proof is by induction over $k \in \{0, 1, \ldots, \ell\}$.

$k = 0$ For this case, note that $\mu = x$, $\mu' = x'$ and if $\eta$ is a path expression then $\eta = \varepsilon$. So, the double implication at 2 is obvious. Let us see when $\eta$ is a node expression. Necessarily $\eta = \langle \varepsilon, \varepsilon, \ldots, \varepsilon \rangle_*$ with $* \in \mathcal{R} \cup \overline{\mathcal{R}}$, and since $(x, x') \in Z_0$ we have that for all $* \in \mathcal{R} \cup \overline{\mathcal{R}}$, $R_*(x, x, \ldots, x)$ iff $R'_*(x', x', \ldots, x')$. It is the same that $(\mathcal{T}, x) \models \langle \varepsilon, \varepsilon, \ldots, \varepsilon \rangle_*$, if and only if, $(\mathcal{T}', x') \models \langle \varepsilon, \varepsilon, \ldots, \varepsilon \rangle_*$.

$k > 0$ Suppose $\eta$ is a path expression with $0 < \text{dd}(\eta) \leq k$ then $\eta =\downarrow \xi\gamma$ where $\xi$ is an empty string or a node test, and $\gamma$ is an empty string or a normal path expression. Note that $\text{dd}(\xi\gamma) \leq k-1$. Let us see the most interesting case when $\xi$ and $\gamma$ are not the empty string, so $\xi = [\varphi]$ for some normal node expression $\varphi$. Suppose $\mu = (x \downarrow y) \odot \nu$ and $\mu' = (x' \downarrow y') \odot \nu'$ with $\nu \in \text{Path}_{k-1}(\mathcal{T}, y)$ and $\nu' \in \text{Path}_{k-1}(\mathcal{T}, y')$. If $(\mathcal{T}, \mu) \models \eta$ then $(\mathcal{T}, y) \models \varphi$ and $(\mathcal{T}, \nu) \models \gamma$. By induction hypothesis, $(\mathcal{T}', y') \models \varphi$ and $(\mathcal{T}', \nu') \models \gamma$, and therefore $(\mathcal{T}', \mu') \models \eta$. Similarly, we can see that if $(\mathcal{T}', \mu') \models \eta$ then $(\mathcal{T}, \mu) \models \eta$.

Suppose now $\eta$ is a node expression and for simplicity, of the form $\eta = \langle\downarrow [\varphi_1]\alpha_1, \downarrow [\varphi_2]\alpha_2\rangle_*$ where $* \in \mathcal{R} \cup \overline{\mathcal{R}}$, $\varphi_1, \varphi_2$ are normal node expressions and $\alpha_1, \alpha_2$ are normal path expressions non-equal to $\varepsilon$ (the rest of the cases are proved by the same idea). Let $x$ be a node in $T$ and $x'$ a node in $T'$ such that $(x, x') \in Z_k$. If $(\mathcal{T}, x) \models \eta$ then there are paths $\mu_1, \mu_2 \in \text{Path}_k(\mathcal{T}, x)$ such that $(\mathcal{T}, \mu_1) \models \downarrow [\varphi_1]\alpha_1$, $(\mathcal{T}, \mu_2) \models \downarrow [\varphi_2]\alpha_2$ and $R_*(\text{end}(\mu_1), \text{end}(\mu_2))$. Since $(x, x') \in Z_k$, by Zig, for the paths $\mu_1$ and $\mu_2$ there are another paths $\mu_1', \mu_2' \in \text{Path}_k(\mathcal{T}', x')$ such that for all $j = 1, 2$ we have that $\text{len}(\mu_j) = \text{len}(\mu_j')$, if $i \in \{0, 1, \ldots, \text{len}(\mu_j)\}$ then $([\mu_j]_i, [\mu_j']_i) \in Z_{k-i}$, and $R_*(\text{end}(\mu_1), \text{end}(\mu_2))$ iff $R_*'(\text{end}(\mu_1'), \text{end}(\mu_2'))$. As $\varphi_1, \varphi_2, \alpha_1, \alpha_2$ have downward depth at most $k-1$, we can apply induction hypothesis and conclude that for all $j \in \{1, 2\}$ if $\mu_j = (x \downarrow y_j) \odot \nu_j$ for some $\nu_j \in \text{Path}_{k-1}(\mathcal{T}, y_j)$ and $\mu_j' = (x' \downarrow y_j') \odot \nu_j'$ for some $\nu_j' \in \text{Path}_{k-1}(\mathcal{T}', y_j')$ then $(\mathcal{T}, y_j) \models \varphi_j$ iff $(\mathcal{T}', y_j') \models \varphi_j$, and $(\mathcal{T}, \nu_j) \models \alpha_j$ iff $(\mathcal{T}', \nu_j') \models \alpha_j$. Thus, $(\mathcal{T}', x') \models \eta$. Similarly (by Zag), $(\mathcal{T}', x') \models \eta$ implies $(\mathcal{T}, x) \models \eta$.

**Proposition 2.** Given $(\mathcal{T}, x)$ and $(\mathcal{T}', x')$ we have that if $(\mathcal{T}, x) \leftrightharpoons_\ell^{\mathcal{R}} (\mathcal{T}', x')$ then $(\mathcal{T}, x) \equiv_\ell^{\mathcal{R}} (\mathcal{T}', x')$.

*Proof.* Suppose $(\mathcal{T}, x) \leftrightharpoons_\ell^{\mathcal{R}} (\mathcal{T}', x')$ via $\mathcal{Z} = \{Z_0, Z_1, \ldots, Z_\ell\}$ with $(x, x') \in Z_\ell$. We want to see that for all $\varphi \in \text{XPath}_{\mathcal{R}}^{\ell}(\downarrow)$ we have that $(\mathcal{T}, x) \models \varphi$ iff $(\mathcal{T}', x') \models \varphi$. It suffices to show this for all normal path test $\varphi = \langle\alpha_1, \alpha_2, \ldots, \alpha_n\rangle_*$ with $* \in \mathcal{R} \cup \overline{\mathcal{R}}$: if $(\mathcal{T}, x) \models \varphi$ then there are paths $\mu_1, \mu_2, \ldots, \mu_n \in \text{Path}_\ell(\mathcal{T}, x)$ such that $(\mathcal{T}, \mu_i) \models \alpha_i$ for all $i \in \{1, 2, \ldots, n\}$ and $R_*(\text{end}(\mu_1), \text{end}(\mu_2), \ldots, \text{end}(\mu_n))$. Because $(x, x') \in Z_\ell$, there are paths $\mu_1', \mu_2', \ldots, \mu_n' \in \text{Path}_\ell(\mathcal{T}', x')$ satisfying Zig with respect to the paths $\mu_1, \mu_2, \ldots, \mu_n$. Thus, by Lemma 4, if $i \in \{1, 2, \ldots, n\}$ then $(\mathcal{T}, \mu_i) \models \alpha_i$ iff $(\mathcal{T}', \mu_i') \models \alpha_i$. Since $R_*(\text{end}(\mu_1), \text{end}(\mu_2), \ldots, \text{end}(\mu_n))$, if and only if, $R_*'(\text{end}(\mu_1'), \text{end}(\mu_2'), \ldots, \text{end}(\mu_n'))$, we conclude that $(\mathcal{T}', x') \models \varphi$ via the paths $\mu_1', \mu_2', \ldots, \mu_n'$. Similarly, by Zag, if $(\mathcal{T}', x') \models \varphi$ then $(\mathcal{T}, x) \models \varphi$.

**Proposition 3.** Given $(\mathcal{T}, x)$ and $(\mathcal{T}', x')$ we have that if $(\mathcal{T}, x) \equiv_\ell^{\mathcal{R}} (\mathcal{T}', x')$ then $(\mathcal{T}, x) \leftrightharpoons_\ell^{\mathcal{R}} (\mathcal{T}', x')$.

*Proof.* First we define for all $k \in \{0, 1, \ldots, \ell\}$ the relations $(z, z') \in Z_k$ iff $(\mathcal{T}, z) \equiv_k^{\mathcal{R}} (\mathcal{T}', z')$. We want to see that the family $\mathcal{Z} := \{Z_0, Z_1, \ldots, Z_\ell\}$ is a $\mathcal{R}_\ell$-bisimulation, and since by hypothesis $(x, x') \in Z_\ell$, we will have that $(\mathcal{T}, x) \leftrightharpoons_\ell^{\mathcal{R}} (\mathcal{T}', x')$ via $\mathcal{Z}$.

Given $Z_k \in \mathcal{Z}$ and $(z, z') \in Z_k$, we want to show that the Zig and Zag clauses are satisfied. Let us see only Zag (for Zig, the same idea works).

Let $(\mu'_1, \mu'_2, \ldots, \mu'_n)$ be a sequence of paths in $\text{Path}_k(\mathcal{T}, z')$ and $* \in \mathcal{R} \cup \overline{\mathcal{R}}$ such that $R'_*(\text{end}(\mu'_1), \ldots, \text{end}(\mu'_n))$, we consider the set

$$P := \{(\mu_1, \mu_2, \ldots, \mu_n) \in \text{Path}_k(\mathcal{T}, z)^n \mid \forall i \in \{1, 2, \ldots, n\} \\ \text{len}(\mu_i) = \text{len}(\mu'_i) \wedge R_*(\text{end}(\mu_1), \ldots, \text{end}(\mu_n))\}$$

Note that $P \neq \emptyset$ because since $(z, z') \in Z_k$

$$(\mathcal{T}', z') \models \langle \downarrow^{\text{len}(\mu'_1)}, \downarrow^{\text{len}(\mu'_2)}, \ldots, \downarrow^{\text{len}(\mu'_n)} \rangle_* \in \text{XPath}^k_{\mathcal{R}}(\downarrow) \\ \Leftrightarrow (\mathcal{T}, z) \models \langle \downarrow^{\text{len}(\mu'_1)}, \downarrow^{\text{len}(\mu'_2)}, \ldots, \downarrow^{\text{len}(\mu'_n)} \rangle_*$$

where $\downarrow^N$ is an abbreviation for the concatenation of $N$ symbols $\downarrow$, and for convenience $\downarrow^0 := \varepsilon$.

Suppose Zag does not hold for the paths $\mu'_1, \mu'_2, \ldots, \mu'_n$. Thus, we have that for all $\overline{\mu} = (\mu_1, \mu_2, \ldots, \mu_n) \in P$ there must be some $\mu_j$ and some $[\mu_j]_i$ such that $([\mu_j]_i, [\mu'_j]_i) \notin Z_{k-i}$.

1. Fix $\overline{\mu} = (\mu_1, \mu_2, \ldots, \mu_n) \in P$. We define a family of node expressions $\{\varphi^{(j,i)}_{\overline{\mu}}\}$ as follows, where $j \in \{1, 2, \ldots, n\}$ and $i \in \{0, 1, \ldots, \text{len}(\mu'_j)\}$: if $(j_0, i_0)$ is the smallest pair (by the lexicographic order) with $([\mu_{j_0}]_{i_0}, [\mu'_{j_0}]_{i_0}) \notin Z_{k-i_0}$ then there exists some node expression $\psi \in \text{XPath}^{k-i_0}_{\mathcal{R}}(\downarrow)$ such that $(\mathcal{T}, [\mu_{j_0}]_{i_0}) \not\models \psi$ and $(\mathcal{T}', [\mu'_{j_0}]_{i_0}) \models \psi$. So, let $\varphi^{(j_0,i_0)}_{\overline{\mu}}$ be equal to such $\psi$, and for the rest of the pairs $(j, i)$ let $\varphi^{(j,i)}_{\overline{\mu}} := \top$ where $\top$ is some fixed tautological node expression with downward depth zero (such as that from Remark 1).
2. Now, for each $j \in \{1, 2, \ldots, n\}$ and $i \in \{0, 1, \ldots, \text{len}(\mu'_j)\}$ let $\Phi^{(j,i)}$ be a formula such that for every $(\widetilde{\mathcal{T}}, \tilde{x})$, $(\widetilde{\mathcal{T}}, \tilde{x}) \models \Phi^{(j,i)}$ iff for all $\varphi^{(j,i)}_{\overline{\mu}}$ with $\overline{\mu} \in P$ we have $(\widetilde{\mathcal{T}}, \tilde{x}) \models \varphi^{(j,i)}_{\overline{\mu}}$ (informally, abusing the notation in the case that $P$ is infinite $\Phi^{(j,i)} \equiv \bigwedge_{\overline{\mu} \in P} \varphi^{(j,i)}_{\overline{\mu}}$). This formula exists because the expressions $\varphi^{(j,i)}_{\overline{\mu}}$ have downward depth at most $k - i$, and thus by Lemma 1 they are finite modulo equivalence. Thus $\Phi^{(j,i)}$ can be defined as the conjunction of some witnesses from each of the (finite) equivalence classes.
3. Finally, for each $j \in \{1, 2, \ldots, n\}$ consider

$$\alpha_j := [\Phi^{(j,0)}] \downarrow [\Phi^{(j,1)}] \downarrow \ldots \downarrow [\Phi^{(j,\text{len}(\mu'_j))}]$$

By construction, $\text{dd}(\alpha_j) \leq k$ and $(\mathcal{T}', \mu'_j) \models \alpha_j$. Therefore, $(\mathcal{T}', x') \models \langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle_*$ and since $(\mathcal{T}, x) \equiv^{\mathcal{R}}_k (\mathcal{T}', x')$, it must be that $(\mathcal{T}, x) \models \langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle_*$. But, this means that there are paths $\mu_1, \ldots, \mu_n \in \text{Path}_k(\mathcal{T}, x)$ satisfying $(\mathcal{T}, \mu_j) \models \alpha_j$ and $R_*(\text{end}(\mu_1), \ldots, \text{end}(\mu_n))$. As $\mu := (\mu_1, \ldots, \mu_n) \in P$, there must be some $\mu_j$ with $(\mathcal{T}, \mu_j) \not\models \alpha_j$. So, we obtain a contradiction by assuming that Zag is not satisfied.

**Definition 16.** *Let* $\boldsymbol{\sigma_{\mathcal{R}} := \{F_{\downarrow}\} \cup \{L_r\}_{r\in\mathcal{R}}}$ *be a signature where* $F_{\downarrow}$ *is binary and for each* $r \in \mathcal{R}$ *the symbol* $L_r$ *is* $\mathcal{A}(r)$*-ary. We interpret the symbols in* $\sigma_{\mathcal{R}}$ *in an* $\mathcal{R}$*-tree* $\mathcal{T} = \langle T, E_{\downarrow}, \{R_r\}_{r\in\mathcal{R}}\rangle$ *as* $[\![F_{\downarrow}]\!]_{\mathcal{T}} := E_{\downarrow}$ *and* $[\![L_r]\!]_{\mathcal{T}} := R_r$ *for* $r \in \mathcal{R}$*.*

**Definition 17.** *If* $FO(\sigma_{\mathcal{R}})$ *is the set of first order formulas with signature* $\sigma_{\mathcal{R}}$ *and equality, we define recursively the truth-preserving* **standard translation** $ST(-)$ *from* $XPath_{\mathcal{R}}(\downarrow)$ *to* $FO(\sigma_{\mathcal{R}})$ *which maps node expressions to formulas with one free variable and path expression to formulas with two free variables.*

$$ST(\varphi \wedge \psi)(u) := ST(\varphi)(u) \wedge ST(\psi)(u) \qquad ST(\neg\varphi)(u) := \neg ST(\varphi)(u)$$
$$ST(\langle\alpha_1, \alpha_2, \ldots, \alpha_n\rangle_r)(u) := (\exists v_1, \ldots, v_n)\left(\textstyle\bigwedge_{i=1}^{n} ST(\alpha_i)(u, v_i) \wedge L_r(v_1, \ldots, v_n)\right)$$
$$\text{for } r \in \mathcal{R}, \text{ with } \mathcal{A}(r) = n$$
$$ST(\langle\alpha_1, \alpha_2, \ldots, \alpha_n\rangle_{\overline{r}})(u) := (\exists v_1, \ldots, v_n)\left(\textstyle\bigwedge_{i=1}^{n} ST(\alpha_i)(u, v_i) \wedge \neg L_r(v_1, \ldots, v_n)\right)$$
$$\text{for } r \in \mathcal{R}, \text{ with } \mathcal{A}(r) = n$$
$$ST(\varepsilon)(u, v) := (u = v) \qquad ST(\downarrow)(u, v) := F_{\downarrow}(u, v)$$
$$ST([\varphi])(u, v) := (u = v) \wedge ST(\varphi)(u)$$
$$ST(\alpha\beta)(u, v) := (\exists w)(ST(\alpha)(u, w) \wedge ST(\beta)(w, v))$$

**Lemma 2.** For each $\leftrightarrows^{\mathcal{R}}$-invariant $\psi(u) \in \mathrm{FO}(\sigma_{\mathcal{R}})$ with one free variable $u$, there is $\ell \geq 0$ such that $\psi$ is $\ell$-local.

*Proof.* We follow the Step 1 of the proof of van Benthem/Rosen's theorem for the modal logic given by Martin Otto in [8].

Let $\ell := 2^q - 1 \geq 0$ where $q$ is the quantifier rank of $\psi(u)$. We want to show that for an arbitrary pointed model $(\mathcal{T}, x)$, $\mathcal{T} \models \psi[u \mapsto x]$ iff $\mathcal{T}|_{\ell}^{x} \models \psi[u \mapsto x]$, and hence, $\psi(u)$ is $\ell$-local.

Given a pointed model $(\mathcal{T}, x)$ we define two new $\mathcal{R}$-trees $\mathcal{T}_A$ and $\mathcal{T}_B$ as follows:

- The underlying tree of $\mathcal{T}_A$ is constructed by tying from a fresh node $t_A$: the original $(\mathcal{T}, x)$, $q$-copies of $(\mathcal{T}, x)$ via a family $\mathcal{I}$ of $q$ isomorphisms (as $\sigma_{\mathcal{R}}$-structures) from $\mathcal{T}$, and $q$-copies of $(\mathcal{T}|_{\ell}^{x}, x)$ via a family $\mathcal{J}$ of $q$ isomorphisms from $\mathcal{T}|_{\ell}^{x}$. For each $r \in \mathcal{R}$ of arity $n$, the relation $\tilde{R}_r$ in $\mathcal{T}_A$ is defined by extending the relation $R_r$ in $\mathcal{T}$ as $\tilde{R}_r(y_1, y_2, \ldots, y_n)$ iff $R_r(x_1, x_2, \ldots, x_n)$ where for all $k \in \{1, 2, \ldots, q\}$ there exists $f \in \mathcal{I} \cup \mathcal{J}$ such that $f(x_k) = y_k$. Notice that $t_A$ doesn't play a relevant role because it is not related to any of the nodes, even itself.
- $\mathcal{T}_B$ and $t_B$ are constructed almost exactly like $\mathcal{T}_A$ and $t_A$, but now we replace the original $(\mathcal{T}, x)$ with $(\mathcal{T}|_{\ell}^{x}, x)$.

Clearly, $(\mathcal{T}_A, x) \leftrightarrows^{\mathcal{R}} (\mathcal{T}, x)$ and $(\mathcal{T}_B, x) \leftrightarrows^{\mathcal{R}} (\mathcal{T}|_{\ell}^{x}, x)$. Suppose $\mathcal{T} \models \psi[u \mapsto x]$, and since $\psi(u)$ is $\leftrightarrows^{\mathcal{R}}$-invariant we have that $\mathcal{T}_A \models \psi[u \mapsto x]$. We affirm that $(\mathcal{T}_A, x) \equiv_q (\mathcal{T}_B, x)$ as $\sigma_{\mathcal{R}}$-structures, i.e., they satisfy the same formulas with quantifier rank less than or equal to $q$. To see that, one can follow the idea given by Otto showing a winning strategy for player **II** in a Ehrenfeucht-Fraïssé game.

Therefore, as $\psi(u)$ has quantifier rank $q$, $\mathcal{T}_B \models \psi[u \mapsto x]$, and then $\mathcal{T}|_{\ell}^{x} \models \psi[u \mapsto x]$.

## References

1. Clark, J., DeRose, S.: XML path language (XPath). Website (1999). W3C Recommendation
2. Gottlob, G., Koch, C., Pichler, R.: Efficient algorithms for processing XPath queries. TODS **30**, 444–491 (2005)
3. Bojańczyk, M., Muscholl, A., Schwentick, T., Segoufin, L.: Two-variable logic on data trees and XML reasoning. JACM **56**, 1–48 (2009)
4. Figueira, D., Figueira, S., Areces, C.: Model theory of XPath on data trees. Part I: bisimulation and characterization. JAIR **53**, 271–314 (2015)
5. Abriola, S., Descotte, M.E., Figueira, S.: Model theory of XPath on data trees. Part II: binary bisimulation and definability. Inf. Comput. **255**, 195–223 (2017)
6. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press, Cambridge (2001)
7. van Benthem, J.: Modal correspondence theory. Ph.D. thesis, Universiteit van Amsterdam (1976)
8. Otto, M.: Elementary proof of the van Benthem-Rosen characterisation theorem. Technical report 2342, Fachbereich Mathematik, Technische Universität Darmstadt (2004)