Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 328 (2016) 3–16

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

On Regular Paths with Counting and Data Tests

Everardo Bárcenas[1]

CONACYT
Universidad Veracruzana
Xalapa, Veracruz, Mexico

Edgard Benítez-Guerrero[2]

Facultad de Estadística e Informática
Universidad Veracruzana
Xalapa, Veracruz, Mexico

Jesús Lavalle[3]

Benemérita Universidad Autónoma de Puebla
Instituto Nacional de Astrofísica, Óptica y Electrónica
Puebla, Puebla, Mexico

Abstract

Regular path expressions represent the navigation core of the XPath query language for semi-structured data (XML), and it has been characterized as the First Order Logic with Two Variables (FO^2). Data tests refers to (dis)equality comparisons on data tree models, which are unranked trees with two kinds of labels, propositions from a finite alphabet, and data values from a possibly infinite alphabet. Node occurrences on tree models can be constrained by counting/arithmetic constructors. In this paper, we identify an EXPTIME extension of regular paths with data tests and counting operators. This extension is characterized in terms of a closed under negation Presburger tree logic. As a consequence, the EXPTIME bound also applies for standard query reasoning (emptiness, containment and equivalence).

Keywords: Modal Logics, XPath, Automated Reasoning, Data Trees, Counting

1 Introduction

XPath is a W3C standard query language for semi-structured data (XML), and it also takes an important role in many XML technologies, such as, XProc, XSLT,

[1] iebarcenaspa@conacyt.mx

[2] edbenitez@uv.mx

[3] jlavalle@cs.buap.mx

http://dx.doi.org/10.1016/j.entcs.2016.11.002

and XQuery [19,2]. The navigation core of XPath, also known as regular path queries, has been recently characterized by the First Order Logic of Two Variables (FO^2) [19]. Models for this logic are unranked trees, where nodes are labeled by propositions from a finite alphabet. Data tests, also known as data joins in databases community, on XPath queries are expressions of the forms $\rho_1 \equiv \rho_2$ and $\rho_1 \not\equiv \rho_2$. These expressions hold whenever data values (propositions from an infinite alphabet) contained in path ρ_1 are (dis)equal to data values contained in path ρ_2, respectively. Another important constructors on XPath queries concerns counting: $\rho_1 \# \rho_2$, where $\# \in \{\leq, >, =, \neq\}$. These expressions hold whenever the number of nodes denoted by ρ_1 and ρ_2 satisfies constraint $\#$. There are several recent works studying regular path extensions with either data tests or counting [14,4,13,12,2]. However, as far as we know, the current work represent the first study on regular path extensions concerning both constructors, data tests and counting. More precisely, we give a characterization of a regular paths with data test with respect to constants $\rho \equiv k$ ($\rho \not\equiv k$) and with counting operators on children paths. For this characterization we use a modal tree logic equipped with a fixed point operator, converse modalities and Presburger arithimetic constraints [3]. Due to this characterization, the EXPTIME bound from the logic is imported to standard query reasoning (emptiness, containment, and equivalence) with counting and data tests.

1.1 Related works

There are several extensions of FO^2 with data tests [16,8,6,7]. In [16], $FO^2(<, +1, \equiv)$ for data trees is introduced: $<$ stands for descendants and following sibling relations, $+1$ refers to child and next sibling relations, and $\equiv$ is a binary predicate for data tests. Decidability, without any complexity analysis, for $FO^2(<, +1, \equiv)$ in data trees is first shown by a reduction to the reachability problem of a counter tree automata model. Previously in [6], the same result was obtained for data words (one branched tree), more precisely, $FO^2(<, +1, \equiv)$ for data words was shown decidable by a reduction to reachability problem of Petri nets. A 3NEXPTIME upper bound for $FO^2(<, +1, \equiv)$ is implied in [6]. Even earlier in [7], $FO^2(+1, \equiv)$ for trees was introduced and shown decidable also in 3NEXPTIME. Regarding model theoretic results of extensions of FO with data models, in [8] several cases are studied: data words, data trees and data graphs. In another direction, regarding regular paths (XPath navigation core), it is well know data test on full navigation regular paths is undecidable [13]. Several fragments (downward, forward, transitive) of regular path expressions with data tests are studied [11,15,12,13,14]. With their corresponding complexity ranging from EXPTIME to non elementary. Contrastingly, in this paper, instead of restricting navigation on queries, we study the full navigation (children, parents, following and previous sibling, descendants and ancestros) regular path expressions, but we restrict data tests to constants only, that is, expressions of the form $\rho \equiv k$ ($\rho \not\equiv k$).

Regarding regular paths with counting, there are several recent studies [19,2,4]. In [19], it was show the extension of regular paths with counting is in general undecidable. EXPTIME fragments (counting with respect to constants) were later identified in [2,4]. In [2], it is shown regular paths with counting operators on children paths with respect to constants $\rho > k$ is decidable in EXPTIME. The same bound was later shown for the extension of regular paths with counting on full navigation paths, also with respect to constants. In the current work, we study another counting extension $\rho_1 \# \rho_2$ ($\# \in \{\leq, >, =, \neq\}$), where expressions ρ_1 and ρ_2 are restricted to children paths. Several other logics with counting have been proposed in the setting of unranked trees [9,18,10,3,17]. A fragment of ambient logic is shown decidable via a reduction to the Presburger modal logic (modal logik K + Presburger arithmetic) in [9]. Presburger modal logic was later shown to be in PSPACE-complete [10]. In [18], en extension of Presburger modal logic with a fixed point is shown decidable in EXPTIME. The same bound was shown in [3] for a further extension of the logic with converse modalities and nominals (K+Presburger arithmetic+fixed point+converse+nominals). We show in the current paper this Presburger logic can actually characterize regular paths counting on children and data tests (w.r.t. constants). In [17], the EXPTIME bound was further developed for a set of coalegebraic modal logics via a type elimination algorithm. Excepting [5], where the emptiness problem for ranked tree automaton with equality and counting constraints was shown decidable without a further complexity analysis, all the above works study separately data tests and counting. In the current work, we identify an EXPTIME extension of regular paths with both, data tests and counting.

1.2 Outline

In Section 2, we introduce some preliminaries, more precisely, we introduce the notions of trees and data trees. We later describe a counting and data tests extension of regular paths in Section 3. In Section 4, we describe a modal tree logic with a fixed point, converse modalities and Presburger arithmetic constructors. The main result of this paper, which is a characterization of the regular path extension with counting and data tests in terms of the logic, is described in Section 5. Since the characterization is polynomial and the logic is closed under negation, the EXPTIME bound for the logic can be imported for reasoning (emptiness, containment and equivalence) on regular paths with counting and data tests. We conclude with a summary of this work, together with a brief discussion of further research perspectives in Section 6.

2 Preliminaries

For the languages described in the current work, unless otherwise stated, we use a fixed alphabet composed by set of propositions $PROPS$ and a a set of modalities $MODS = \{\downarrow, \uparrow, \leftarrow, \rightarrow\}$. Intuitively, propositions are used in tree models to label nodes, and modalities are interpreted as the children$\downarrow$, parent $\uparrow$, right siblings $\leftarrow$, and left siblings $\rightarrow$ relations.

We now introduce the notion of a tree, which can be seen as a tree-shaped Kripke structure (transition system).

Definition 2.1 [Tree] A tree $\mathcal{T}$ is defined as a tuple (N, R, L), such that

- N is a finite set of nodes;
- $R : N \times MODS \times N$ is a transition relation among nodes and modalities forming a tree (we often write $n \in R(n', m)$ instead of $(n', m, n) \in R$); and
- $L : N \times PROPS$ is a left-total labeling relation (we often write $p \in L(n)$ instead of $(n, p) \in L$).

The set of data values are the set of natural numbers $\mathbb{N}$. Data trees can be seen as an extension of trees (Definition 2.1), where nodes are labeled with data values and propositions.

Definition 2.2 [Data tree] A data tree Γ is defined as a tuple (N, R, L, D), such that

- (N, R, L) is a tree; and
- $D : N \mapsto \mathbb{N}$ is a total function.

3 Regular Path Queries with Counting and Data Tests

We now describe regular path expressions [19] extended with counting and data tests. Basic path expressions have the form $\alpha : p$, where the axis α can be the children $\downarrow$, the parent $\uparrow$, the right siblings $\leftarrow$, the left siblings $\rightarrow$, the descendants $\downarrow^\star$, and the ancestors $\uparrow^\star$. For instance, $\downarrow^\star\!: p$ selects descendants labeled with proposition p. Paths ρ_1 and ρ_2 can also be composed ρ_1/ρ_2. Consider for example expression $\downarrow^\star\!: p/\downarrow\!: q$. This expression navigates to p descendants, and from there, it selects the q children. Also, paths can also be filtered by qualifiers. Expression $\downarrow^\star\!: p[\downarrow\!: q]$ selects the p descendants with at least one q children. Other arithmetic constraints can be expressed by qualifiers, but only on children paths. For instance, $\downarrow^\star\!: p[\downarrow\!: p_1 =\downarrow\!: p_2]$ selects the p descendants with the same number of children named p_1 and p_2. Data tests also appear as qualifiers. For example, $\downarrow^\star\!: p[\downarrow\!: q = 5]$ selects p descendants with at least one q child with data value equal to integer 5.

We now give a precise syntax of regular paths with counting and data tests.

Definition 3.1 [Syntax] We define the RPQCD expressions (queries) by the following grammar:

$$\begin{aligned}\rho &:= \top \mid \alpha \mid p \mid \alpha : p \mid \rho/\rho \mid \rho[\beta]\\ \beta &:= \rho \mid \rho - \rho \,\#\, k \mid \rho \equiv k \mid \neg\beta \mid \beta \vee \beta\end{aligned}$$

where $p \in PROPS$, $k \in \mathbb{N}$, $\# \in \{>, \leq, =\}$ and $\alpha \in \{\downarrow, \uparrow, \leftarrow, \rightarrow, \downarrow^\star, \uparrow^\star\}$.

In the case of $\rho_1 - \rho_2 \,\#\, k$, both ρ_i $(i = 1, 2)$ are restricted to be children paths, that is, they have one of the following forms: $\downarrow$, $\downarrow\!: p$, $\downarrow [\beta]$ or $\downarrow\!: p[\beta]$.

RPQCD expressions are interpreted over data trees: $\top$ selects the entire set of nodes; $\alpha : p$ navigates through α and selects the p nodes; ρ_1/ρ_2 is the compositions of paths; and $\rho[\beta]$ selects the nodes denoted by ρ satisfying condition β. In particular, when β is $\rho \equiv k$, it holds whenever there is a node denoted by ρ whose data value is equal to k. $\rho_1 - \rho_2 \# k$ is true if and only if the number of nodes selected by ρ_1 minus the number of nodes selected by ρ_2, satisfies constraint $\#k$. Notice some syntactic sugar (notation) as $\rho_1 \# \rho_2$ instead of $\rho_1 - \rho_2 \# 0$ can also be defined. Negation and disjunction are interpreted as expected.

We now give a precise description on how RPQCD expressions are interpreted over data trees.

Definition 3.2 [Semantics] Given a data tree $\Gamma = (N, R, L, D)$, RPQCD expressions are interpreted as follows:

$$\begin{aligned}
[\![\top]\!]^\Gamma &= N \times N \\
[\![p]\!]^\Gamma &= \{(n,n) \mid p \in L(n)\} \\
[\![\alpha]\!]^\Gamma &= \left\{(n_1,n_2) \mid n_1 \overset{\alpha}{\to} n_2\right\} \\
[\![\alpha : p]\!]^\Gamma &= \left\{(n_1,n_2) \in [\![\alpha]\!]^\Gamma \mid p \in L(n_2)\right\} \\
[\![\rho_1/\rho_2]\!]^\Gamma &= [\![\rho_1]\!]^\Gamma \circ [\![\rho_2]\!]^\Gamma \\
[\![\rho[\beta]]\!]^\Gamma &= \left\{(n_1,n_2) \in [\![\rho]\!]^\Gamma \mid n_2 \in [\![\beta]\!]^\Gamma\right\} \\
[\![\rho]\!]^\Gamma &= \left\{n \mid (n,n') \in [\![\rho]\!]^\Gamma\right\} \\
[\![\rho_1 - \rho_2 \# k]\!]^\Gamma &= \left\{n \mid \left|\left\{n_1 \mid (n,n_1) \in [\![\rho_1]\!]^\Gamma\right\}\right| - \left|\left\{n_2 \mid (n,n_2) \in [\![\rho_2]\!]^\Gamma\right\}\right| \# k\right\} \\
[\![\rho \equiv k]\!]^\Gamma &= \left\{n \mid (n',n) \in [\![\rho]\!]^\Gamma, D(n) = k\right\} \\
[\![\neg\beta]\!]^\Gamma &= N \setminus [\![\beta]\!]^\Gamma \\
[\![\beta_1 \vee \beta_2]\!]^\Gamma &= [\![\beta_1]\!]^\Gamma \cup [\![\beta_2]\!]^\Gamma
\end{aligned}$$

where $n_1 \overset{\alpha}{\to} n_2$ holds, if and only if, n_1 is related to n_2 through α in Γ.

We also interpret RPQCD expressions with respect to a context, more precisely, the interpretation of a RPQCD expression ρ on a data tree Γ from a subset of nodes N' (of Γ) is defined as follows:

$$[\![\rho]\!]^\Gamma_{N'} = \left\{n' \mid (n,n') \in [\![\rho]\!]^\Gamma, n \in N'\right\}$$

We now define the standard query reasoning problems for RPQCD: emptiness, containment and equivalence.

Definition 3.3 [Reasoning]

- We say a RPQCD expression ρ is empty, if and only if, for any data tree Γ, we have that $[\![\rho]\!]^\Gamma \neq \emptyset$.
- Given two RPQCD expressions ρ_1 and ρ_2, we say ρ_1 is contained in ρ_2, written $\rho_1 \subseteq \rho_2$, if and only if, for any data tree Γ, we have that $[\![\rho_1]\!]^\Gamma \subseteq [\![\rho_2]\!]^\Gamma$.
- Given two RPQCD expressions ρ_1 and ρ_2, we say ρ_1 is equivalent to ρ_2, if and only if, $\rho_1 \subseteq \rho_2$ and $\rho_2 \subseteq \rho_1$.

4 A Presburger Tree Logic

We now describe a modal tree logic, as originally introduced in [3], with a fixed point, converse modalities and Presburger arithmetic operators.

Definition 4.1 [Syntax] We inductively define the set of μTLIC formulas by the following grammar:

$$\phi := p \mid \neg\phi \mid \phi \vee \phi \mid \langle m \rangle \phi \mid \mu x.\phi \mid \phi - \phi \# k$$

where $p \in PROPS$, $m \in MODS$, $\# \in \{>, \leq, =\}$, and $k \in \mathbb{N}$ coded in binary form.

μTLIC expressions are interpreted as subset tree nodes: propositions are used as node labels; negation is interpreted as set complement; disjunction as set union; modal formulas $\langle m \rangle \phi$ holds in nodes where there is at least one m transition to a node supporting ϕ; the fixed point operator $\mu x.\phi$ is interpreted as a recursion operator; and Presburger formulas $\phi - \psi \# k$ selects nodes whose ϕ children minus ψ children satisfy constraint $\# k$.

Before formally introduce the interpretation of μTLIC formulas, we first define a valuation function $V : X \mapsto N$ of set of variables x over a set of nodes of a given tree.

Definition 4.2 [Semantics] Given a tree $\mathcal{T} = (N, R, L)$ and a valuation V, μTLIC formulas are interpreted as follows:

$$\begin{aligned}
[\![p]\!]_V^{\mathcal{T}} &= \{n \mid p \in L(n)\} \\
[\![\neg\phi]\!]_V^{\mathcal{T}} &= N \setminus [\![\phi]\!]_V^{\mathcal{T}} \\
[\![\phi \vee \psi]\!]_V^{\mathcal{T}} &= [\![\phi]\!]_V^{\mathcal{T}} \cup [\![\psi]\!]_V^{\mathcal{T}} \\
[\![\langle m \rangle \phi]\!]_V^{\mathcal{T}} &= \left\{n \mid R(n, m) \cap [\![\phi]\!]_V^{\mathcal{T}}\right\} \\
[\![\mu x.\phi]\!]_V^{\mathcal{T}} &= \bigcap \left\{M \mid [\![\phi]\!]_{V[^M/_x]}^{\mathcal{T}} \subseteq M\right\} \\
[\![\phi - \phi \# k]\!]_V^{\mathcal{T}} &= \left\{n \mid \left|R(n, \downarrow) \cap [\![\phi]\!]_V^{\mathcal{T}}\right| - \left|R(n, \downarrow) \cap [\![\psi]\!]_V^{\mathcal{T}}\right| \# k\right\}
\end{aligned}$$

Without loss of generality, we assume variables can only occur bounded, and in the scope of modal or counting formulas [3]. Furthermore equivalent negated normal forms can also be achieved by traditional De Morgan's and modal rules:

$$\begin{aligned}
\neg \langle m \rangle \phi &:= [m] \neg\phi \\
\neg(\phi \vee \psi) &:= \neg\phi \wedge \neg\psi \\
\neg\mu x.\phi &:= \nu x.\neg\phi \left[^x/_{\neg x}\right] \\
\neg(\phi - \psi > k) &:= \phi - \psi \leq k \\
\neg(\phi - \psi \leq k) &:= \phi - \psi > k \\
\neg(\phi - \psi = k) &:= \phi - \psi \neq k \\
\neg(\phi - \psi \neq k) &:= \phi - \psi = k
\end{aligned}$$

We conclude this Section recalling the complexity of μTLIC.

Theorem 4.3 ([3]) *μTLIC is in EXPTIME-complete.*

5 Logic characterization

In this Section we give a characterization of RPQCD expressions in terms of μTLIC formulas.

First we define a non-data version of data trees. Intuitively, data values in data trees are represented by children nodes labeled by a fresh proposition δ. For instance, if a node has value k, then its non-data version has k children labeled by δ. Then, Presburger formulas can be used to test values in non-data trees.

Definition 5.1 Provided a data tree $\Gamma = (N, R, L, D)$, we define the tree $\mathcal{T}(\Gamma) = (N', R', L')$ as follows:

- let N_i be a set of k_i new nodes ($N \cap N_i = \emptyset$) induced by data values of nodes in N, that is, for each $n_i \in N$, $D(n_i) = k_i$, then $N' = N \cup \bigcup_{i=1}^{|N|} N_i$;
- let $R_i = \{n_i\} \times \{\downarrow\} \times N_i$, then $R' = R \cup \bigcup_{i=1}^{|N|} R_i$;
- and let $L_i : N_i \times \{\delta\}$ be left total, then $L' = L \cup \bigcup_{i=1}^{|N|} L_i$, provided δ is a proposition not occurring in L, that is, for each $n \in N$, if $(n, p) \in L$, then $\delta \neq p$.

Example 5.2 Consider for instance the following data tree Γ:

- $N = \{n_0, n_1, n_2, n_3, n_4\}$;
- $n_1 \in R(n_0, \downarrow)$, $n_2, n_3, n_4 \in R(n_1, \downarrow)$;
- $p \in L(n_1)$, $q \in L(n_2)$, $q \in L(n_3)$ and $r \in L(n_4)$; and
- $D(n_1) = 3$, $D(n_i) = 0$ $(i = 0, 2, 3, 4)$.

The corresponding tree $\mathcal{T}(\Gamma)$ is then defined as follows:

- $N = \{n_0, n_1, n_2, n_3, n_4, n_5, n_6, n_7\}$;
- $n_1 \in R(n_0, \downarrow)$, $n_2, n_3, n_4, n_5, n_6, n_7 \in R(n_1, \downarrow)$; and
- $p \in L(n_1)$, $q \in L(n_2)$, $q \in L(n_3)$, $r \in L(n_4)$, and $\delta \in L(n_i)$ $(i = 5, 6, 7)$.

In Figure 1, it is depicted a graphical representation of Γ and $\mathcal{T}(\Gamma)$.

We now give a precise translation of regular paths with counting and data tests in terms of the logic.

Definition 5.3 We define a translation function F of RPQCD expressions in terms

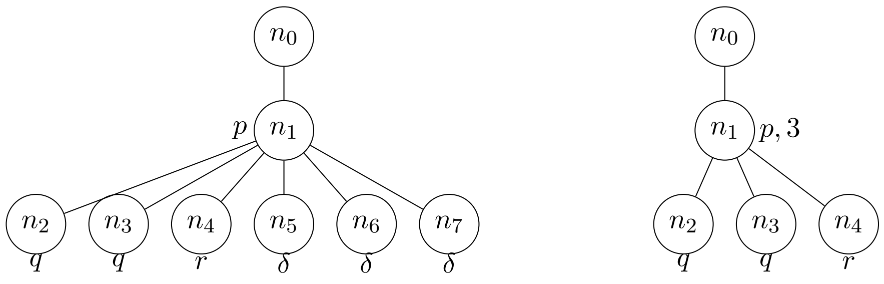

Fig. 1. A data tree Γ on the right with its corresponding tree $\mathcal{T}(\Gamma)$ on the left.

of the logic as follows:

$$\begin{array}{ll}
F(\top, C) := C \wedge \neg\delta & F(p, C) := p \wedge \neg\delta \wedge C \\
F(\downarrow, C) := \neg\delta \wedge \langle\uparrow\rangle\, C & F(\uparrow, C) := \neg\delta \wedge \langle\downarrow\rangle\, C \\
F(\leftarrow, C) := \neg\delta \wedge \langle\rightarrow\rangle\, C & F(\rightarrow, C) := \neg\delta \wedge \langle\leftarrow\rangle\, C \\
F(\downarrow^\star, C) := \neg\delta \wedge \mu x. \langle\uparrow\rangle\,(C \vee x) & F(\uparrow^\star, C) := \neg\delta \wedge \mu x. \langle\downarrow\rangle\,(C \vee x) \\
F(\alpha : p, C) := F(\alpha, C) \wedge F(p, \top) & F(\rho_1/\rho_2, C) := F(\rho_2, F(\rho_1, C)) \\
F(\rho[\beta], C) := F(\rho, C) \wedge G(\beta, \top) &
\end{array}$$

where δ is a fresh proposition and G is a translation of qualifiers (Definition 5.4).

Definition 5.4 We define a translation of qualifiers in terms of the logic as follows:

$$\begin{array}{ll}
G(\top, C) := C \wedge \neg\delta & G(\alpha, C) := F(\overline{\alpha}, C) \\
G(p, C) := p \wedge \neg\delta \wedge C & G(\alpha : p, C) := F(\overline{\alpha}, C \wedge p \wedge \neg\delta) \\
G(\rho_1/\rho_2, C) := G(\rho_1, G(\rho_2, C)) & G(\rho[\beta], C) := G(\rho, G(\beta, \top) \wedge C) \\
G(\neg\beta, C) := \neg G(\beta, C) & G(\beta_1 \vee \beta_2, C) := G(\beta_1, C) \vee G(\beta_2, C) \\
G(\rho \equiv k, C) := G^{\equiv}(\rho \equiv k, C) & G(\rho_1 - \rho_2 \,\#\, k, C) := G^{\#}(\rho_1, C) - G^{\#}(\rho_2, C) \,\#\, k
\end{array}$$

$$\begin{aligned}
G^{\equiv}(\top \equiv k, C) &:= (\phi^\delta \wedge \langle\uparrow\rangle\, G(\top, C) = k \\
G^{\equiv}(p \equiv k, C) &:= (\phi^\delta \wedge \langle\uparrow\rangle\, G(p, C)) = k \\
G^{\equiv}(\alpha \equiv k, C) &:= G(\alpha, (\phi^\delta \wedge \langle\uparrow\rangle\, C) = k) \\
G^{\equiv}(\alpha : p \equiv k, C) &:= G(\alpha, (\phi^\delta \wedge \langle\uparrow\rangle\,(C \wedge p)) = k) \\
G^{\equiv}(\rho_1/\rho_2 \equiv k, C) &:= G(\rho_1, G^{\equiv}(\rho_2 \equiv k, C)) \\
G^{\equiv}(\rho[\beta] \equiv k, C) &:= G^{\equiv}(\rho \equiv k, G(\beta, \top) \wedge C) \\
\phi^\delta &:= \delta \wedge \neg p' \wedge \neg \langle\downarrow\rangle\, \top
\end{aligned}$$

$$\begin{array}{ll}
G^{\#}(\downarrow, C) := C \wedge \neg\delta & G^{\#}(\downarrow : p, C) := p \wedge \neg\delta \wedge C \\
G^{\#}(\downarrow [\beta], C) := G(\beta, \top) \wedge C & G^{\#}(\downarrow : p[\beta], C) := p \wedge \neg\delta \wedge G(\beta, \top) \wedge C
\end{array}$$

provided that $\overline{\alpha}$ is the dual relation of α, more precisely, $\overline{\downarrow} = \uparrow$, $\overline{\leftarrow} = \rightarrow$, $\overline{\downarrow^\star} = \uparrow^\star$, and $\overline{\overline{\alpha}} = \alpha$; and where p' represents all other propositions distinct to δ (recall the set of propositions is finite).

Example 5.5 Consider expression $\downarrow\colon p$ evaluated from an arbitrary context $\top$, then its corresponding translation is $F(\downarrow\colon p, \top) := \neg\delta \wedge \langle\uparrow\rangle \top \wedge p$. In the data tree Γ defined in Example 5.2 and depicted in Figure 1, it is easy to see $\downarrow\colon p$ selects n_1, whereas $F(\downarrow\colon p, \top)$ also selects n_1 but in the corresponding tree $\mathcal{T}(\Gamma)$.

Consider now $\downarrow\colon p[\downarrow\colon q > \downarrow\colon r]$ which is the short form of $\downarrow\colon p[\downarrow\colon q - \downarrow\colon r > 0]$. This query selects the p children with more q children than r ones. See again Example 5.2 and Figure 1 for a data tree model. The corresponding translation is then

$$F(\downarrow\colon p[\downarrow\colon q > \downarrow\colon r], \top) := \neg\delta \wedge \langle\uparrow\rangle \top \wedge p \wedge ((q \wedge \neg\delta) > (r \wedge \neg\delta))$$

We now consider an example with a data test $\downarrow\colon p[\top \equiv 3]$. This expression denotes p children with data value equals to 3. The corresponding translation is then computed as follows:

$$F(\downarrow\colon p[\top \equiv 3]) := \neg\delta \wedge \langle\uparrow\rangle \top \wedge p \wedge (\delta \wedge \neg p \wedge \neg \langle\downarrow\rangle \top \wedge \langle\uparrow\rangle \top) = 3$$

In Example 5.2 and in Figure 1 there is an instance model of the corresponding data tree Γ and tree $\Gamma(\mathcal{T})$.

Since translation of paths consider a context represented by formulas, we now give a non-data version of formulas. Intuitively, context formulas are indistinguishably interpreted over data and non-data trees.

Definition 5.6 [Context formula] Given a formula ϕ in negated normal form, its corresponding context formula ϕ^C is inductively defined as follows:

$$\begin{array}{ll}
p^C := p & (\neg p)^C := \neg\delta \wedge \neg p \\
(\phi \vee \psi)^C := \phi^C \vee \psi^C & (\phi \wedge \psi)^C := \phi^C \wedge \psi^C \\
(\langle m\rangle \phi)^C := \neg\delta \wedge \langle m\rangle \phi^C & ([m]\, \phi)^C := \neg\delta \wedge [m]\, \phi^C \\
(\mu x.\phi)^C := \left(\phi \left[{}^{\mu x.\phi}/_{x}\right]\right)^C & (\nu x.\phi)^C := \left(\phi \left[{}^{\nu x.\phi}/_{x}\right]\right)^C \\
(\phi - \psi \,\#\, k)^C := \phi^C - \psi^C \,\#\, k &
\end{array}$$

Lemma 5.7 *Given any data tree Γ, for any formula ϕ and any valuation V, we have that*

$$[[\phi^C]]^{\Gamma}_{V} = [[\phi^C]]^{\mathcal{T}(\Gamma)}_{V}$$

Proof. By structural induction on ϕ.

The base cases where the input formula is a proposition or the negation of proposition is straightforward by construction of $\mathcal{T}(\mathcal{T})$ (Definition 5.1). Notice nodes where δ is true does not support any other proposition.

The case of disjunction and conjunction are immediate by induction.

If the input formula is a modal formula $\langle m \rangle \psi$ or $[m] \psi$, it is also immediate by induction and by noticing nodes where δ is true are always leaves.

For fixed point cases $\mu x.\psi$ or $\nu x.\psi$, we apply a second induction on the expansions $\psi \left[{}^{\mu x.\psi}/_{x}\right]$ or $\psi \left[{}^{\nu x.\psi}/_{x}\right]$, which are guaranteed to be equivalent by the Fixed Point Theorem [3]. This second induction is immediate since variables always occur under the scope of a modality or a counting operator.

The case of counting formulas $\psi - \varphi \# k$ is also immediate by induction. □

We now describe the main result of this paper: a sound characterization of regular paths with counting and data tests in terms of Presburger formulas.

Theorem 5.8 (Logic characterization of data queries) *For any ρ RPQCD expression, data tree Γ, μTLIC context formula ϕ^C, and any valuation V, we have the following:*

- $[\![\rho]\!]^{\Gamma}_{[\![\phi^C]\!]^{\mathcal{T}(\Gamma)}_V} = [\![F\left(\rho, \phi^C\right)]\!]^{\mathcal{T}(\Gamma)}_V$*; and*
- $F\left(\rho, \phi^C\right)$ *is of polynomial size with respect to q and ϕ^C.*

Proof. For the first item, we now proceed by induction on the input expression ρ.

Base cases where the input expression is either $\top$ or p are immediate by Lemma 5.7.

Consider now the case of $\alpha : p$, say α is $\downarrow^\star$. Then

$$F\left(\downarrow^\star\!: p, \phi^C\right) := p \wedge \neg\delta \wedge \mu x. \langle\uparrow\rangle \left(\phi^C \vee x\right)$$

Since nodes where δ holds are always leaves, then $\neg\delta \wedge \mu x. \langle\uparrow\rangle\, \phi^C \vee x$ consistently characterizes descendants $(\downarrow^\star)$ of ϕ^C. Other base cases are analogous.

For the inductive step, consider now ρ_1/ρ_2. Then

$$F\left(\rho_1/\rho_2, \phi^C\right) := F\left(\rho_2, F\left(\rho_1, \phi^C\right)\right)$$

By induction we know that

$$[\![\rho_1]\!]^{\mathcal{T}}_{[\![\phi^C]\!]^{\mathcal{T}(\Gamma)}_V} = [\![F\left(\rho_1, \phi^C\right)]\!]^{\mathcal{T}(\Gamma)}_V$$
$$[\![\rho_2]\!]^{\mathcal{T}}_{[\![F(\rho_1,\phi^C)]\!]^{\mathcal{T}(\Gamma)}_V} = [\![F\left(\rho_2, F\left(\rho_1, \phi^C\right)\right)]\!]^{\mathcal{T}(\Gamma)}_V$$

then, if we compose ρ_1 and ρ_2, we obtain

$$[\![\rho_1/\rho_2]\!]^{\mathcal{T}}_{[\![\phi^C]\!]^{\mathcal{T}(\Gamma)}_V} = [\![F\left(\rho_2, F\left(\rho_1, \phi^C\right)\right)]\!]^{\mathcal{T}(\Gamma)}_V$$

Consider now the case $\rho[\beta]$. We know

$$F\left(\rho[\beta], \phi^C\right) := F\left(\rho, \phi^C\right) \wedge G(\beta, \top)$$

By induction we know $F\left(\rho, \phi^C\right)$ corresponds to ρ, that is,

$$[\![\rho]\!]^{\Gamma}_{[\![\phi^C]\!]^{\mathcal{T}(\Gamma)}_V} = [\![F\left(\rho, \phi^C\right)]\!]^{\mathcal{T}(\Gamma)}_V$$

If we show β corresponds to $G(\beta,\top)$, that is, $\llbracket\beta\rrbracket^{\Gamma} = \llbracket G(\beta,\top)\rrbracket_V^{\mathcal{T}(\Gamma)}$, we can clearly then infer the full correspondence of the input expression, that is,

$$\llbracket\rho[\beta]\rrbracket^{\Gamma}_{\llbracket\phi^C\rrbracket_V^{\mathcal{T}(\Gamma)}} = \left\llbracket F\left(\rho,\phi^C\right)\right\rrbracket_V^{\mathcal{T}(\Gamma)} \cap \llbracket G(\beta,\top)\rrbracket_V^{\mathcal{T}(\Gamma)}$$

Consider first the case where β is $\rho[\rho_1-\rho_2\#k]$. Now, recall that

$$G(\rho_1-\rho_2\#k,\top) := \Big(G^{\#}(\rho_1,\top) - G^{\#}(\rho_2,\top)\Big)\,\#k$$

We now proceed by a second induction on the structure of ρ_i $(i=1,2)$. Assume now ρ_i are $\downarrow\colon p_i$, then

$$G^{\#}(\rho_1,\top) := p_i \wedge \neg\delta \wedge \top$$

Recall nodes where δ holds does not support any other proposition, hence $G(\downarrow\colon p_1-\downarrow\colon p_2\#k,\top)$ clearly counts p_i children only, and holds where the corresponding counted children satisfy constraint $\#k$, that is,

$$\llbracket\downarrow\colon p_1-\downarrow\colon p_2\,\#\,k\rrbracket^{\Gamma} = \llbracket G(\downarrow\colon p_1-\downarrow\colon p_2\,\#\,k,\top)\rrbracket_V^{\mathcal{T}(\Gamma)}$$

The other base cases, where one of ρ_i have the form $\downarrow$ and the other ρ_j is $\downarrow\colon p$, are analogous.

Consider now ρ_i are $\downarrow\colon p_i[\beta_i]$. The corresponding formula is then

$$G^{\#}(\downarrow\colon p_i[\beta_i],\top) = p_i \wedge \neg\delta \wedge \top \wedge G(\beta_i,\top)$$

In this case we apply yet another induction on β_i to show its correspondence with $G(\beta,\top)$. This induction case is identical to the case of the first induction. It is then clear the correspondence of the arithmetical expression

$$\llbracket\downarrow\colon p_1[\beta_1]-\downarrow\colon p_2[\beta_2]\,\#\,k\rrbracket^{\Gamma} = \llbracket G(\downarrow\colon p_1[\beta_1]-\downarrow\colon p_2[\beta_2]\,\#\,k,\top)\rrbracket^{\mathcal{T}(\Gamma)}$$

For the other inductive cases, where one of ρ_i has the forms $\downarrow$ or $\downarrow\colon p_i$ and the other ρ_j is $\downarrow\colon p_j[\beta_j]$, we proceed similarly.

We now consider the case when β has the form $\rho[\rho_0\equiv k]$. Another induction is applied on ρ_0. Assume ρ_0 have the form $\downarrow^{\star}\colon p$, then

$$\begin{aligned} G(\downarrow^{\star}\colon p\equiv k,\top) &:= G(\alpha,(\phi^{\delta}\wedge G(p,\top)) = k \\ &:= \mu x.\, \langle\downarrow\rangle\, \big((\delta\wedge\neg p'\wedge\neg\langle\downarrow\rangle\top\wedge\langle\uparrow\rangle(p\wedge\neg\delta)) = k \vee x\big) \end{aligned}$$

This formula navigates recursively through children (descendants), until it finds nodes with k δ children. These children do not support other propositions p', they are leaves $\neg\langle\downarrow\rangle\top$, and its parent is labeled by p. Then clearly

$$\llbracket\downarrow^{\star}\colon p\equiv k\rrbracket^{\Gamma} = \llbracket G(\downarrow^{\star}\colon p\equiv k,\top)\rrbracket_V^{\mathcal{T}(\Gamma)}$$

Other base cases are proven analogously.

When ρ_0 is ρ_1/ρ_2, then

$$G(\rho_1/\rho_2 \equiv k, \top) := G(\rho_1, G^{\equiv}(\rho_2 \equiv k, \top))$$

By induction, we know $G(\rho_1)$ corresponds to nodes selected by ρ_1, that is,

$$[\![\rho_1]\!]^{\mathcal{T}} = [\![F(\rho_1, \top)]\!]_V^{\mathcal{T}(\Gamma)}$$

Also by induction, we know $G(\rho_2 \equiv k, \top)$ selects the same nodes than ρ_2 with k δ children. Hence if we evaluate ρ_1 from the $\rho_2 \equiv k$ nodes, we thus obtain

$$[\![\rho_1/\rho_2 \equiv k]\!]^{\Gamma} = [\![G(\rho_1/\rho_2 \equiv k, \top)]\!]_V^{\mathcal{T}(\Gamma)}$$

Consider now ρ_0 has the form $\rho[\beta]$.

$$G(\rho[\beta] \equiv k, \top) := G^{\equiv}(\rho \equiv k, \top) \wedge G(\beta, \top)$$

By induction we know $G^{\equiv}(\rho \equiv k, \top)$ corresponds to $\rho \equiv k$:

$$[\![\rho \equiv k]\!]^{\Gamma} = [\![G(\rho \equiv k, \top)]\!]_V^{\mathcal{T}(\Gamma)}$$

We now proceed by another induction on β to show $G(\beta, \top)$ corresponds β: $[\![\beta]\!]^{\Gamma} = [\![G(\beta, \top)]\!]_V^{\mathcal{T}(\Gamma)}$ This induction is identical as the one of the first inductive step. Hence, $G(\rho[\beta] \equiv k, \top)$ precisely selects the ρ nodes satisfying β $(\rho[\beta])$ with k δ children:

$$[\![\rho[\beta] \equiv k]\!]^{\Gamma} = [\![\rho[\beta] \equiv k]\!]_V^{\mathcal{T}(\Gamma)}$$

In the second inductive step, now consider when β is a negation $\neg\beta_0$. This case goes smoothly by induction by noticing negation on counting and data test expressions are safe, that is, $\neg G(\rho_1 - \rho_2 = k)$ is the same than $G(\rho_1 - \rho_2 \neq k)$, and vice-versa, analogously $\leq$ and $\equiv$ are the dual of $>$ and $\not\equiv$, respectively.

Finally, disjunction $\beta_1 \vee \beta_2$ is immediate by induction.

Regarding the translation size, it is easy to see the resulting formula is of polynomial size by noticing the translation function does not introduce duplications, excepting p', introduced by data tests. □

An immediate consequence of Theorems 4.3 and 5.8 is an EXPTIME bound for RPQCD reasoning.

Corollary 5.9 *Reasoning (emptiness, containment and equivalence) on regular path queries with counting and data tests (RPQCD) is in EXPTIME.*

6 Discussion

We introduced an extension of regular path expressions with counting and data tests. Counting operators express occurrence restrictions on children path expressions, whereas data tests express (dis)equality relations among paths with respect to their data values. We give a characterization of the extension of regular paths in terms of a Presburger logic originally introduced in [3]. Since the characterization is polynomial and the logic is closed under negation, the EXPTIME bound of the logic is then imported for the emptiness, containment and equivalence of paths with counting and data tests. As a first further research perspective we propose the study of the model checking problem of the Presburger logic (it is known a quadratic-time model checking algorithm for the logic without converse modalities [1]). This would imply complexity bound for the query evaluation of paths with counting and data tests. As another future work, we propose the study of further data test extensions of regular paths, in the setting of expressive modal logics with efficient reasoning Fischer-Ladner algorithms as in [3].

References

[1] Bárcenas, E., E. Benítez-Guerrero and J. Lavalle, *On the model checking of the graded μ-calculus on trees*, in: *14th Mexican International Conference on Artificial Intelligence*, 2015, pp. 178–189.

[2] Bárcenas, E., P. Genevès, N. Layaïda and A. Schmitt, *Query reasoning on trees with types, interleaving, and counting*, in: T. Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011* (2011), pp. 718–723.
URL http://ijcai.org/papers11/Papers/IJCAI11-127.pdf

[3] Bárcenas, E. and J. Lavalle, *Expressive reasoning on tree structures: Recursion, inverse programs, Presburger constraints and nominals*, in: *12th Mexican International Conference on Artificial Intelligence*, 2013, pp. 80–91.

[4] Bárcenas, E. and J. Lavalle, *Global numerical constraints on trees*, Logical Methods in Computer Science **10** (2014).
URL http://dx.doi.org/10.2168/LMCS-10(2:10)2014

[5] Barguñó, L., C. Creus, G. Godoy, F. Jacquemard and C. Vacher, *Decidable classes of tree automata mixing local and global constraints modulo flat theories*, Logical Methods in Computer Science **9** (2013).
URL http://dx.doi.org/10.2168/LMCS-9(2:1)2013

[6] Bojańczyk, M., C. David, A. Muscholl, T. Schwentick and L. Segoufin, *Two-variable logic on data words*, ACM Trans. Comput. Log. **12** (2011), p. 27.
URL http://doi.acm.org/10.1145/1970398.1970403

[7] Bojańczyk, M., A. Muscholl, T. Schwentick and L. Segoufin, *Two-variable logic on data trees and XML reasoning*, J. ACM **56** (2009).
URL http://doi.acm.org/10.1145/1516512.1516515

[8] Bojańczyk, M. and T. Place, *Toward model theory with data values*, in: A. Czumaj, K. Mehlhorn, A. M. Pitts and R. Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part II*, Lecture Notes in Computer Science **7392** (2012), pp. 116–127.
URL http://dx.doi.org/10.1007/978-3-642-31585-5_14

[9] Dal-Zilio, S., D. Lugiez and C. Meyssonnier, *A logic you can count on*, in: N. D. Jones and X. Leroy, editors, *Proceedings of the 31st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2004, Venice, Italy, January 14-16, 2004* (2004), pp. 135–146.
URL http://doi.acm.org/10.1145/964001.964013

[10] Demri, S. and D. Lugiez, *Complexity of modal logics with presburger constraints*, J. Applied Logic **8** (2010), pp. 233–252.
URL http://dx.doi.org/10.1016/j.jal.2010.03.001

[11] Figueira, D., *Forward-xpath and extended register automata on data-trees*, in: L. Segoufin, editor, *Database Theory - ICDT 2010, 13th International Conference, Lausanne, Switzerland, March 23-25, 2010, Proceedings*, ACM International Conference Proceeding Series (2010), pp. 231–241.
URL http://doi.acm.org/10.1145/1804669.1804699

[12] Figueira, D., *Decidability of downward xpath*, ACM Trans. Comput. Log. **13** (2012), p. 34.
URL http://doi.acm.org/10.1145/2362355.2362362

[13] Figueira, D., *On xpath with transitive axes and data tests*, in: R. Hull and W. Fan, editors, *Proceedings of the 32nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2013, New York, NY, USA - June 22 - 27, 2013* (2013), pp. 249–260.
URL http://doi.acm.org/10.1145/2463664.2463675

[14] Figueira, D., S. Figueira and C. Areces, *Model theory of xpath on data trees. part I: bisimulation and characterization*, J. Artif. Intell. Res. (JAIR) **53** (2015), pp. 271–314.
URL http://dx.doi.org/10.1613/jair.4658

[15] Figueira, D. and L. Segoufin, *Bottom-up automata on data trees and vertical xpath*, in: T. Schwentick and C. Dürr, editors, *28th International Symposium on Theoretical Aspects of Computer Science, STACS 2011, March 10-12, 2011, Dortmund, Germany*, LIPIcs **9** (2011), pp. 93–104.
URL http://dx.doi.org/10.4230/LIPIcs.STACS.2011.93

[16] Jacquemard, F., L. Segoufin and J. Dimino, *Fo2(<, +1, ˜) on data trees, data tree automata and branching vector addition systems*, Logical Methods in Computer Science **12** (2016).
URL http://dx.doi.org/10.2168/LMCS-12(2:3)2016

[17] Kupke, C., D. Pattinson and L. Schröder, *Reasoning with global assumptions in arithmetic modal logics*, in: A. Kosowski and I. Walukiewicz, editors, *Fundamentals of Computation Theory - 20th International Symposium, FCT 2015, Gdańsk, Poland, August 17-19, 2015, Proceedings*, Lecture Notes in Computer Science **9210** (2015), pp. 367–380.
URL http://dx.doi.org/10.1007/978-3-319-22177-9_28

[18] Seidl, H., T. Schwentick and A. Muscholl, *Counting in trees*, in: J. Flum, E. Grädel and T. Wilke, editors, *Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas].*, Texts in Logic and Games **2** (2008), pp. 575–612.

[19] ten Cate, B., T. Litak and M. Marx, *Complete axiomatizations for xpath fragments*, J. Applied Logic **8** (2010), pp. 153–172.
URL http://dx.doi.org/10.1016/j.jal.2009.09.002