Query Reasoning on Data Trees with Counting

Everardo Bárcenas[1,2], Edgard Benítez-Guerrero[2], and Jesús Lavalle[3,4]

[1] CONACYT
[2] Universidad Veracruzana
[3] Benemérita Universidad Autónoma de Puebla
[4] Instituto Nacional de Astrofísica, Óptica y Electrónica
iebarcenaspa@conacyt.mx,edbenitez@uv.mx,jlavalle@cs.buap.mx

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).

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, and XQuery [1, 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) [1]. 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 [3–6, 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 [7]. 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.

There are several extensions of FO^2 with data tests [8–11]. In [8], $FO^2(<,+1,\equiv)$ 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 [10], the same result was obtained for data words (one branched tree). Even earlier in [11], $FO^2(+1,\equiv)$ for trees was introduced and shown decidable in 3NEXPTIME. In another direction, regarding regular paths (XPath navigation core), it is well know data test on full navigation regular paths is undecidable [5]. Several fragments (downward, forward, transitive) of regular path expressions with data tests are studied [12, 13, 6, 5, 3]. 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.

Regarding regular paths with counting, there are several recent studies [1, 2, 4]. In [1], 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]. Several other logics with counting have been proposed in the setting of unranked trees [14–16, 7, 17]. In [17], the EXPTIME bound was further developed for a set of coalegebraic modal logics via a type elimination algorithm. Excepting [18], 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.

We describe a counting and data tests extension of regular paths in Section 2. In Section 3, 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 4. We conclude with a summary of this work, together with a brief discussion of further research perspectives in Section 5.

2 Regular Path Queries with Counting and Data Tests

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 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 1), where nodes are labeled with data values and propositions.

Definition 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.*

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

Definition 3 (Syntax). *We define the RPQCD expressions (queries) by the following grammar:*

$$\rho := \top \mid \alpha \mid p \mid \alpha : p \mid \rho/\rho \mid \rho[\beta]$$
$$\beta := \rho \mid \rho - \rho \,\#\, k \mid \rho \equiv k \mid \neg\beta \mid \beta \vee \beta$$

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 4 (Semantics). *Given a data tree $\Gamma = (N, R, L, D)$, RPQCD expressions are interpreted as follows:*

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

where $n_1 \xrightarrow{\alpha} n_2$ 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 5 (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 $\rho_2 \subseteq \rho_1$.*

3 A Presburger Tree Logic

We now describe a modal tree logic, as originally introduced in [7], with a fixed point, converse modalities and Presburger arithmetic operators.

Definition 6 (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 7 (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 [7]. Furthermore equivalent negated normal forms can also be achieved by traditional De Morgan's and modal rules: $\neg \langle m \rangle \phi := [m] \neg\phi$, $\neg(\phi \vee \psi) := \neg\phi \wedge \neg\psi$, $\neg\mu x.\phi := \nu x.\neg\phi\, [^x/_{\neg x}]$, $\neg(\phi - \psi > k) := \phi - \psi \leq k$, $\neg(\phi - \psi \leq k) := \phi - \psi > k$, $\neg(\phi - \psi = k) := \phi - \psi \neq k$, and $\neg(\phi - \psi \neq k) := \phi - \psi = k$.

We conclude this Section recalling the complexity of μTLIC.

Theorem 1 ([7]). *μTLIC is in EXPTIME-complete.*

4 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 8. *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* $\delta \neq p$.

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

Definition 9. *We define a translation function* F *of RPQCD expressions in terms 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 10).*

Definition 10. *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).*

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 11 (Context formula). *Given a formula ϕ in negated normal form, its corresponding context formula ϕ^C is inductively defined as follows:*

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

Lemma 1. *Given any data tree Γ, for any formula ϕ and any valuation V, we have that $[\![\phi^C]\!]^{\Gamma}_{V} = [\![\phi^C]\!]^{\mathcal{T}(\Gamma)}_{V}$.*

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

Theorem 2 (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.*

An immediate consequence of Theorems 1 and 2 is an EXPTIME bound for RPQCD reasoning.

Corollary 1. *Reasoning (emptiness, containment and equivalence) on regular path queries with counting and data tests (RPQCD) is in EXPTIME.*

5 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 [7]. 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 [19]). 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 [7].

Acknowledgment. This work was partially developed under the support of the Mexican National Science Council (CONACYT) in the scope of the Cátedras CONACYT project "Infraestructura para Agilizar el Desarrollo de Sistemas Centrados en el Usuario" (Ref 3053).

References

1. ten Cate, B., Litak, T., Marx, M.: Complete axiomatizations for xpath fragments. J. Applied Logic **8**(2) (2010) 153–172
2. Bárcenas, E., Genevès, P., Layaïda, N., Schmitt, A.: Query reasoning on trees with types, interleaving, and counting. In: IJCAI, International Joint Conference on Artificial Intelligence. (2011)
3. Figueira, D., Figueira, S., Areces, C.: Model theory of xpath on data trees. part I: bisimulation and characterization. J. Artif. Intell. Res. (JAIR) **53** (2015) 271–314
4. Bárcenas, E., Lavalle, J.: Global numerical constraints on trees. Logical Methods in Computer Science **10**(2) (2014)
5. Figueira, D.: On xpath with transitive axes and data tests. In: Symposium on Principles of Database Systems, PODS. (2013) 249–260
6. Figueira, D.: Decidability of downward xpath. ACM Trans. Comput. Log. **13**(4) (2012) 34
7. Bárcenas, E., Lavalle, J.: Expressive reasoning on tree structures: Recursion, inverse programs, Presburger constraints and nominals. In: Mexican International Conference on Artificial Intelligence, MICAI. (2013) 80–91
8. Jacquemard, F., Segoufin, L., Dimino, J.: Fo2(<, +1, ˜) on data trees, data tree automata and branching vector addition systems. Logical Methods in Computer Science **12**(2) (2016)
9. Bojańczyk, M., Place, T.: Toward model theory with data values. In: Automata, Languages, and Programming - International Colloquium, ICALP. (2012)
10. Bojańczyk, M., David, C., Muscholl, A., Schwentick, T., Segoufin, L.: Two-variable logic on data words. ACM Trans. Comput. Log. **12**(4) (2011) 27
11. Bojańczyk, M., Muscholl, A., Schwentick, T., Segoufin, L.: Two-variable logic on data trees and XML reasoning. J. ACM **56**(3) (2009)
12. Figueira, D.: Forward-xpath and extended register automata on data-trees. In: Database Theory - ICDT, International Conference. (2010)
13. Figueira, D., Segoufin, L.: Bottom-up automata on data trees and vertical xpath. In: Symposium on Theoretical Aspects of Computer Science, STACS. (2011)
14. Dal-Zilio, S., Lugiez, D., Meyssonnier, C.: A logic you can count on. In: Symposium on Principles of Programming Languages, POPL. (2004)
15. Seidl, H., Schwentick, T., Muscholl, A.: Counting in trees. In: Logic and Automata. (2008)
16. Demri, S., Lugiez, D.: Complexity of modal logics with presburger constraints. J. Applied Logic **8**(3) (2010) 233–252
17. Kupke, C., Pattinson, D., Schröder, L.: Reasoning with global assumptions in arithmetic modal logics. In: Fundamentals of Computation Theory, FCT. (2015)
18. Barguñó, L., Creus, C., Godoy, G., Jacquemard, F., Vacher, C.: Decidable classes of tree automata mixing local and global constraints modulo flat theories. Logical Methods in Computer Science **9**(2) (2013)
19. Bárcenas, E., Benítez-Guerrero, E., Lavalle, J.: On the model checking of the graded μ-calculus on trees. In: Mexican International Conference on Artificial Intelligence, MICAI. (2015)