Containment for queries over trees with attribute value comparisons

Marx, M.; Sherkhonov, E.

DOI
10.1016/j.is.2015.11.003

Publication date
2016

Document Version
Final published version

Published in
Information systems

Information Systems 58 (2016) 1–13

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

Containment for queries over trees with attribute value comparisons ☆

Maarten Marx, Evgeny Sherkhonov *

University of Amsterdam, Science Park 904, 1098XH Amsterdam, Netherlands

ARTICLE INFO

Article history:
Received 22 January 2015
Received in revised form
23 October 2015
Accepted 26 November 2015
Available online 5 December 2015

Keywords:
Tree query languages
Conjunctive queries over trees
Positive XPath
Containment
Attribute

ABSTRACT

Björklund et al. [6] showed that containment for conjunctive queries (CQ) over trees and positive XPath is respectively Π_2^P and coNP-complete. In this article we show that the same problem has the same complexity when we expand these languages with XPath's attribute value comparisons. We show that different restrictions on the domain of attribute values (finite, infinite, dense, discrete) have no impact on the complexity. Making attributes required does have an impact: the problem becomes harder. We also show that containment of tree patterns without the wildcard $*$, which is in PTIME, becomes coNP-hard when adding equality and inequality comparisons.

1. Introduction

In this article we study the containment problem for positive XPath (PosXPath) and conjunctive queries (CQ) interpreted over finite unranked ordered trees with respect to the axes *Child*, *NextSibling*, *Descendant*, NextSibling$^+$ and *Following*. PosXPath is a large fragment of Core XPath [14] which contains all the axes and constructs except negation. Conjunctive queries over trees are an analog of relational conjunctive queries, which correspond to the select-from-where SQL queries in which the where-condition uses only conjunctions of equality comparisons, and are the most widely used query language in practice. A thorough study of the containment problem for CQ over trees has been done in [6]. Their main result is Π_2^P-completeness of the problem. In fact, conjunctive queries can be reformulated as the positive fragment of Core XPath with path intersection. Thus, the Π_2^P hardness result also holds for the containment problem for this fragment. Inspection of the proof in [6] also indicates that the containment for just PosXPath remains in coNP. This extends the result of Miklau and Suciu [19], who showed that containment for tree patterns is coNP-complete.

The query language considered in these previous results ignores attributes. However, in many practical scenarios we deal with data that come from numeric domains, such as real or natural numbers. Thus, it is natural to consider conjunctive queries expanded with attribute value comparisons and study basic static analysis problems such as satisfiability and containment. Such an expansion has been considered for Tree Patterns in [1], where a Π_2^P-completeness result for the containment has been established. However, the hardness proof relies on the construct that allows comparisons of attributes of two different nodes, a feature that is not expressible in Core XPath. As a positive counterpart, a coNP upper bound for containment was shown in the case when comparisons are restricted to either so-called left semi-interval or right

☆ An extended abstract announcing some of the results of this paper was presented at the 16th International Workshop on the Web and Databases (WebDB), New York, June 23, 2013.

* Corresponding to: POSTBUS 94323, 1090 GH, Amsterdam, The Netherlands. Tel.: +31 205258627.

E-mail addresses: maartenmarx@uva.nl (M. Marx), e.sherkhonov@uva.nl (E. Sherkhonov).

http://dx.doi.org/10.1016/j.is.2015.11.003
0306-4379/

semi-interval attribute constraints. For an attribute a and constant c, an attribute constraint $(@_a \mathrm{op} c)$ is left semi-interval if $\mathrm{op} \in \{<, \leq, =\}$.

This article is an extension of [23], where it was shown that the complexity of containment does not increase for tree patterns expanded with *both* left and right semi-intervals constraints together with inequality constraint. Here we show that essentially the complexity does not change in cases of positive XPath and conjunctive queries over trees. Furthermore, the same upper bounds hold for the cases when we make certain assumptions on the underlying attribute domain D. That is, we show that all the complexity results still hold for the cases when D is a dense or discrete infinite linear order, with or without endpoints, or a finite linear order. As another result, we show that by requiring at least one attribute to be defined in every node of a tree, the complexity of containment over such trees rises to PSPACE. If, on the other hand, we require attributes to be defined only at nodes with a certain label (which can be expressed in DTDs) the complexity remains in coNP.

All the upper bound results for both PosXPath and CQ are obtained from a suitable polynomial reduction to the containment problem in $\mathrm{PosXPath}^{\neg^s}$ and $\mathrm{UCQ}^{\neg^s}$ (PosXPath and CQ expanded with safe label negation and union) over trees in which nodes may have multiple labels, respectively. Safe label negation is the construct $p\backslash\{q_1, \ldots, q_n\}$ which denotes p-labelled nodes that are not labelled with any of thelabels $q_1, \ldots, q_n$. Table 1 summarizes our results.

The paper is organized as follows. In Section 2 we briefly mention related work. Section 3 contains all the necessary preliminary notions. Section 4 contains the main results. In particular, in Section 4.1 we show that containment for $\mathrm{UCQ}^{\neg^s}$ and $\mathrm{PosXPath}^{\neg^s}$ is in Π_2^p and coNP respectively. Next in Section 4.2 we consider containment for $\mathrm{CQ}^{@}$ and $\mathrm{PosXPath}^{@}$ and show the same upper bounds by reducing to the previous problem. Then in Section 4.3 we show that the upper bounds of containment do not change in case of some natural restrictions on the attribute domain. Section 4.4 contains lower bounds: containment of tree patterns without wildcard rises from PTIME to coNP when we add equality and inequality comparisons; containment of tree patterns rises from coNP to PSPACE when we add equality and inequality comparisons and interpret them on trees in which at least one attribute is defined at each node (a so-called *required* attribute). We finish with conclusions and future work.

Table 1
Complexity results for containment of Positive XPath and CQ with attribute value comparisons.

	$\mathrm{PosXPath}^{@}$	$\mathrm{CQ}^{@}$
No attributes	coNP [6]	Π_2^p [6]
Optional attributes	coNP (Theorem 2)	Π_2^p (Theorem 2)
Required attributes	PSPACE-hard (Theorem 3)	PSPACE-hard (Theorem 3)

2. Related work

Containment of Conjunctive Queries (CQ) with arithmetic comparisons. The classical result on containment of conjunctive queries over relational databases is its NP-completeness [7]. Later, containment for conjunctive queries expanded with arithmetics comparisons was shown to be Π_2^p-complete [17,26]. In [2], Afrati et al. consider various restrictions on type of comparison operations on either of the two input conjunctive queries with comparisons, as well as on interaction between the comparisons, so that the containment is in NP (cf. Table 1 in [2]). However, it was left open what the exact complexity of containment for CQ with comparisons of type $X \mathrm{op} c$ is, where c is a constant and $\mathrm{op} \in \{=, \neq, \leq, \geq, <, >\}$, i.e., the type of comparisons that we consider in this paper. Note that the Π_2^p-lower bound proof in [26] uses disequation of variables, i.e., the construct $X \neq Y$ for variables X and Y. Nevertheless, adding comparisons of the form $X \mathrm{op} c$ to conjunctive queries does change the complexity of containment, which is in contrast with the result of the current paper for PosXPath and CQ over trees. This is argued in [13], where Π_2^p-hardness of containment for CQ with comparisons was shown, using comparisons of the form $X \neq c$. This proof can also be adapted to use comparisons of the form both $X \leq c$ and $X > c$ [21].

Relational conjunctive queries with negated atoms were also studied previously. It is known that containment for CQ with negated atoms is Π_2^p-complete [25]. The analog of safe negation that we consider here was also considered in the context of relational CQ [27]. In this case, negation in a conjunctive query is *safe* if every variable appearing in a negated atom also occurs in a positive atom of the query. Interestingly, the lower bound proof from [13] can be adapted to show Π_2^p-hardness of CQ with safe negation [21]. Thus, adding safe negation to relational conjunctive queries does change the complexity of containment (from NP to Π_2^p), which is in contrast with the result of the current paper where safe negation does not change the complexity of containment for queries over trees.

Containment for queries interpreted over trees. The containment problem for various XPath fragments has been a topic of wide interest for the past several years. A polynomial time algorithm for tree patterns without the wildcard based on homomorphism between queries was given in [3]. The main result of Miklau and Suciu [19] is the coNP-completeness of containment of tree patterns *with* the wildcard. Almost a complete picture of the containment problem for the XPath fragments with disjunction, in the presence of DTDs and variables was given in [20]. Notably, it was shown that with a finite alphabet the containment problem rises to PSPACE. [28] gives decidability results for various fragments with DTDs and a class of integrity constraints. XPath containment in the presence of dependency constraints was studied in [10,11]. All these complexity results are given for forward fragments of XPath. In this paper we consider all the backward axes (parent and ancestor) together with the document order axis (next sibling, following sibling, and following axes and their inverses). Note that in [22] it was shown that every

XPath expression has an equivalent expression without backward axes. However, this translation may result in an expression of exponential size.

A closely related problem is XPath satisfiability [16,4]. [4] contains almost a complete picture of the satisfiability problem with or without the presence of constraints for various fragments of XPath. Query containment reduces to XPath satisfiability in fragments with enough expressive power (e.g., with negation and filter expressions).

Recently the containment of Boolean combinations of tree patterns interpreted over data trees was studied in [9], where Π_2^P-completeness was shown. Their results are incomparable with our results since tree patterns used in [9] output a tuple of data values of nodes, while our tree patterns output nodes.

A closely related work is by Afrati et al. [1]. Consider the containment problem for tree patterns with general arithmetic comparisons. They add the ability to compare the value of an attribute in two different nodes (note that this is not expressible in Core XPath) and show that containment for this fragment is Π_2^P-complete. As mentioned in the introduction, we extend their coNP result for tree patterns with attribute value comparisons.

A systematic study of conjunctive queries interpreted over trees started in [15], where the central problem was the evaluation problem. The authors established a PTime and NP dichotomy of the problem. The containment problem for this language was considered in [6], where it was shown to be Π_2^P-complete. The Π_2^P upper bound was shown via the small counterexample property, similar to the one in [19]. On the other hand, the Π_2^P lower bound proof heavily relies on the DAG structure of conjunctive queries. In fact, if one disallows path intersections and allows disjunction, the same "small counterexample" technique will yield a coNP algorithm. Since XPath expressions with backward axes can be expressed in this language, this implies the coNP upper bound for positive XPath. Containment of conjunctive queries under schema constraints was studied in [5], where 2ExpTime-completeness of the problem was established.

We end with some results on tractable (PTime) containment. As mentioned above, [3] provides a PTime algorithm for containment of tree patterns without the wildcard. PTime containment for acyclic conjunctive queries implies tractability for containment of tree patterns without descendant. Moreover, containment for tree patterns without filters is in PTime as well [19]. However adding attribute value comparisons may raise the complexity. For instance, as shown below in Proposition 5, containment for tree patterns without the wildcard together with equality and inequality attribute comparisons is coNP-hard. As for conjunctive queries interpreted over trees, very limited fragments have PTime containment: those that use one of *Child* or *NextSibling* relation only [6].

3. Preliminaries

We work with node-labelled ordered unranked finite trees, where the nodes are labeled by finite subsets of the infinite set of labels Σ. Formally, a tree over Σ is a tuple $(N, E, <, r, \rho)$, where N, the set of nodes of the tree, is a prefix closed set of finite sequences of natural numbers, $E = \{(\langle n_1, \ldots, n_k\rangle, \langle n_1, \ldots, n_k, n_{k+1}\rangle) | \langle n_1, \ldots, n_{k+1}\rangle \in N\}$ is the child relation, the sibling relation $<$ is defined as $\{(\langle n_1, \ldots, n_k\rangle, \langle n_1, \ldots, n_k+1\rangle) | \langle n_1, \ldots, n_k\rangle, \langle n_1, \ldots, n_k+1\rangle \in N\}$, $r = \langle\rangle$ is the root of the tree, and ρ is the function assigning to each node in N a finite subset of Σ. If for every node n of a tree $\rho(n)$ is singleton, we call such a tree as a *single-labeled* tree. Otherwise, it is *multi-labeled*. A *pointed* tree is a pair T, n, where n is a node in T.

Let A be a set of attribute names and $(D, <)$ a dense linear order without endpoints. Then a tree with attributes from A over Σ is a tuple $(N, E, <, r, \rho, att)$ such that $(N, E, <, r, \rho)$ is a tree over Σ and $att: N \times A \to D$ is a partial function.

By E^+ and $<^+$ we denote the *descendant* and the *following sibling* relations which are transitive closures of the child and sibling relations respectively. We will also use $<_f$ for the *following* relation, i.e., the abbreviation for $(E^{-1})^* \circ <^+ \circ E^*$. For $x, y \in N$ and $R \in \{E, E^+, <, <^+, <_f\}$, by $T \vDash xRy$ we denote the fact that $(x, y) \in R$.

Positive XPath with attribute value comparisons. We define the syntax of Positive XPath (denoted as PosXPath$^{@}$) node and path formulas with attribute value comparisons with the following grammar.

$$step ::= \downarrow \mid \uparrow \mid \leftarrow \mid \rightarrow,$$
$$\varphi ::= p \mid \top \mid @_a \mathrm{op} c \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle \alpha \rangle \varphi,$$
$$\alpha ::= step \mid ?\varphi \mid \alpha; \alpha \mid \alpha \cup \alpha \mid step^+ \mid \rightarrow_f \mid \leftarrow_p,$$

where $p \in \Sigma, a \in A$, $\mathrm{op} \in \{\leq, \geq, <, >, =, \neq\}$, and $c \in D$.

The semantics of PosXPath$^{@}$ path formulas α and node formulas φ is defined as follows. Let $T = (N, E, <, r, \rho, att)$ be a tree over Σ with attribute names from A. In a mutual induction we define the relation $[[\alpha]]_T \subseteq N \times N$ and the satisfaction relation $T, n \vDash \varphi$.

- $[[\downarrow]]_T = E$,
- $[[\uparrow]]_T = E^{-1}$,
- $[[\rightarrow]]_T = <$,
- $[[\leftarrow]]_T = <^{-1}$,
- $[[\rightarrow_f]]_T = <_f$,
- $[[\leftarrow_p]]_T = (<_f)^{-1}$,
- $[[?\varphi]]_T = \{(n, n) \in N \times N \mid T, n \vDash \varphi\}$,
- $[[\alpha; \beta]]_T = [[\alpha]]_T \circ [[\beta]]_T$,
- $[[\alpha \cup \beta]]_T = [[\alpha]]_T \cup [[\beta]]_T$,
- $[[\alpha^+]]_T = ([[\alpha]]_T)^+$ for $\alpha \in \{\downarrow, \uparrow, \rightarrow, \leftarrow\}$,

and

- $T, n \vDash \top$,
- $T, n \vDash p$ iff $p \in \rho(n)$,
- $T, n \vDash @_a \mathrm{op} c$ iff $(D, <) \vDash att(n, a)\ \mathrm{op}\ c$,
- $T, n \vDash \varphi \wedge \psi$ iff $T, n \vDash \varphi$ and $T, n \vDash \psi$,
- $T, n \vDash \varphi \vee \psi$ iff $T, n \vDash \varphi$ or $T, n \vDash \psi$,
- $T, n \vDash \langle \alpha \rangle \varphi$ iff there is a node m with $(n, m) \in [[\alpha]]_T$ and $T, m \vDash \varphi$.

The *step* axes select a pair of nodes that are in the child, parent, next-sibling or previous-sibling relations in the tree. Furthermore, the $\rightarrow_f$ and $\leftarrow_p$ axes select nodes that

are in the following and the preceding relations in the tree respectively. Note that $\top$ is the same as the wildcard axis.

Sometimes we will write $T \vDash \varphi$ to denote $T, r \vDash \varphi$.

Conjunctive queries with attribute value comparisons. Let *Var* be a set of variables, A a set of attribute names and $(D, <)$ the attribute domain, which is a dense linear order without endpoints. A conjunctive query with attribute value comparisons ($\mathsf{CQ}^{@}$) over Σ, A and D is a positive existential first-order formula without disjunction in prenex normal form over a set of unary predicates $p(x)$ and $@_a(x) \circ\!\!\mathrm{p} c$, where $p \in \Sigma$, $x \in Var, c \in D$ and $\circ\!\!\mathrm{p} \in \{\leq, \geq, <, >, =, \neq\}$; and the binary predicates *Child*, *Descendant*, *NextSibling*, *NextSibling*$^+$ and *Following*. If Q is a $\mathsf{CQ}^{@}$, by $Var(Q)$ we denote the set of variables occurring in Q. By $FVar(Q)$ we denote the set of free variables in Q. If $|FVar(Q)| = k > 0$, we call Q a k-ary conjunctive query. If $|FVar(Q)| = 0$, we call Q a Boolean conjunctive query.

Let Q be a conjunctive query and $T = (N, E, <, r, \rho, att)$ a tree over Σ and attributes from A. A *valuation* of Q on T is a total function $\theta: Var(Q) \to N$. A valuation is a *satisfaction* if it satisfies the query, that is, every atom of Q is satisfied by the valuation. Satisfaction of an atom in T, given a valuation θ, is defined as follows.

- $T, \theta \vDash p(x)$ iff $p \in \rho(\theta(x))$,
- $T, \theta \vDash @_a(x) \circ\!\!\mathrm{p} c$ iff $(D, <) \vDash att(\theta(x), a) \circ\!\!\mathrm{p}\ c$,
- $T, \theta \vDash Child(x, y)$ iff $T \vDash \theta(x) E \theta(y)$,
- $T, \theta \vDash Descendant(x, y)$ iff $T \vDash \theta(x) E^+ \theta(y)$,
- $T, \theta \vDash NextSibling(x, y)$ iff $T \vDash \theta(x) < \theta(y)$,
- $T, \theta \vDash NextSibling^+(x, y)$ iff $T \vDash \theta(x) <^+ \theta(y)$,
- $T, \theta \vDash Following(x, y)$ iff $T \vDash \theta(x) <_f \theta(y)$.

A tree T *models* Q, denoted as $T \vDash Q$ if there is a satisfaction of Q on T. If $(x_1, \ldots, x_k)$ is the tuple of free variables in Q, then the *answer* of Q over T is the set $answer(Q, T) = \{(\theta(x_1), \ldots, \theta(x_k)) | \theta$ is a satisfaction of Q on $T\}$. Note that tuples can be nullary as well. Thus, for a Boolean query Q, $answer(Q, T) = \{\langle\rangle\}$ (and we say Q is *true* on T) if there is a satisfaction of Q on T and $answer(Q, T) = \varnothing$ (and we say Q is *false* on T) otherwise.

We also consider unions of conjunctive queries with attribute value comparisons, denoted as $\mathsf{UCQ}^{@}$. These are formulas of the form $\bigvee_{i=1}^{n} Q_i$, where $Q_i \in \mathsf{CQ}^{@}$. The semantics of these formulas is defined in the obvious way.

PosXPath$^{@}$ formulas as CQ$^{@}$ formulas with disjunction. Every PosXPath$^{@}$ formula can be translated into an equivalent $\mathsf{CQ}^{@}$ formula with disjunction in linear time. The translation is a standard translation of XPath into first-order logic language. It is defined by induction on the complexity of path and node formulas of PosXPath$^{@}$ as follows. Note that the translation can be easily modified to yield a translation into the three variable fragment of first order logic.

$TR_{xy}(\downarrow)$	=	$Child(x, y)$
$TR_{xy}(\uparrow)$	=	$Child(y, x)$
$TR_{xy}(\rightarrow)$	=	$NextSibling(x, y)$
$TR_{xy}(\leftarrow)$	=	$NextSibling(y, x)$
$TR_{xy}(?\varphi)$	=	$x = y \wedge TR_x(\varphi)$
$TR_{xy}(\alpha_1; \alpha_2)$	=	$\exists z.(TR_{xz}(\alpha_1) \wedge TR_{zy}(\alpha_2))$ where z is a fresh variable.
$TR_{xy}(\alpha_1 \cup \alpha_2)$	=	$TR_{xy}(\alpha_1) \vee TR_{xy}(\alpha_2)$
$TR_{xy}(\downarrow^+)$	=	$Descendant(x, y)$
$TR_{xy}(\uparrow^+)$	=	$Descendant(y, x)$
$TR_{xy}(\rightarrow^+)$	=	$NextSibling^+(x, y)$
$TR_{xy}(\leftarrow^+)$	=	$NextSibling^+(y, x)$
$TR_{xy}(\rightarrow_f)$	=	$Following(x, y)$
$TR_{xy}(\leftarrow_p)$	=	$Following(y, x)$
$TR_x(p)$	=	$p(x)$
$TR_x(@_a \circ\!\!\mathrm{p} c)$	=	$@_a(x) \circ\!\!\mathrm{p} c$
$TR_x(\top)$	=	$\top$
$TR_x(\varphi_1 \wedge \varphi_2)$	=	$TR_x(\varphi_1) \wedge TR_x(\varphi_2)$
$TR_x(\varphi_1 \vee \varphi_2)$	=	$TR_x(\varphi_1) \vee TR_x(\varphi_2)$
$TR_x(\langle\alpha\rangle\varphi)$	=	$\exists y.(TR_{xy}(\alpha) \wedge TR_y(\varphi))$, where y is a fresh variable.

Query graphs and embeddings. It is convenient to consider $\mathsf{CQ}^{@}$ and PosXPath$^{@}$ without path union and disjunction in the node formulas as graphs [15].

By Σ_A we denote the attribute labels of the form $@_a \circ\!\!\mathrm{p} c$, where $a \in A, c \in D$ and $\circ\!\!\mathrm{p} \in \{\leq, \geq, <, >, =, \neq\}$.

Definition 1 (*Graph query*). Let Q be a $\mathsf{CQ}^{@}$. Then $G_Q = (V, E, E^+, <, <^+, <_f, \rho, \rho_{att})$, where V is the set of nodes, $R \subseteq V \times V$ for $R \in \{E, E^+, <, <^+, <_f\}$, $\rho: V \to 2^{\Sigma}$, $\rho_{att}: V \to 2^{\Sigma_A}$, is a graph query of Q if the following holds.

- $V = Var(Q)$,
- $p \in \rho(x)$ iff $p(x)$ occurs as a conjunct in Q,
- $@_a \circ\!\!\mathrm{p} c \in \rho_{att}(x)$ iff $@_a(x) \circ\!\!\mathrm{p} c$ occurs as a conjunct in Q,
- $(x, y) \in E$ iff $Child(x, y)$ occurs as a conjunct in Q,
- $(x, y) \in E^+$ iff $Descendant(x, y)$ occurs as a conjunct in Q,
- $(x, y) \in <$ iff $NextSibling(x, y)$ occurs as a conjunct in Q,
- $(x, y) \in <^+$ iff $NextSibling^+(x, y)$ occurs as a conjunct in Q,
- $(x, y) \in <_f$ iff $Following(x, y)$ occurs as a conjunct in Q.

By $Nodes(G)$ we denote the set of nodes V of G. We write $G_Q \vDash u_1 R u_2$, to specify that $(u_1, u_2) \in R$ for $R \in \{E, E^+, <, <^+, <_f\}$. Note that for fragments without attribute value comparisons, the value of the labeling function ρ_{att} is always the empty set. In these cases we omit ρ_{att} in query graphs. The semantics of query graphs is given in terms of embeddings, which are essentially valuations for conjunctive queries.

Definition 2 (*Embedding*). Let $T = (N, E, <, r, \rho, att)$ be a tree over Σ with attributes from A and $G = (V, E, E^+, <, <^+, <_f, \rho, \rho_{att})$ a graph query. A function $g: V \to N$ is called an *embedding* of G into T if the following conditions are satisfied.

- Edge preserving. For every $u_1, u_2 \in V$, if $G \vDash u_1 R u_2$ then $T \vDash g(u_1) R g(u_2)$, for any of the edge relations $R \in \{E, E^+, <, <^+, <_f\}$,
- Label preserving. For every $u \in V$, $\rho(u) \subseteq \rho(g(u))$.
- Attribute comparison preserving. For every $u \in V$, if $@_a \circ\!\!\mathrm{p} c \in \rho_{att}(u)$, then $(D, <) \vDash att(g(u), a) \circ\!\!\mathrm{p}\ c$.

Proposition 1. *Let T be a tree, Q a $\mathsf{CQ}^{@}$ query, G_Q its graph query, and θ a function from $Nodes(G_Q)$ to T. Then*

$T, \theta \vDash Q$ iff θ is an embedding of G_Q into T.

Containment. Let Q and P be two k-ary conjunctive queries. We say that P is *contained* in Q, denoted as $P \subseteq Q$, if for every single-labeled tree T, it holds that

$answer(P,T) \subseteq answer(Q,T)$. We also say that P is contained in Q over multi-labeled trees and denote it by $P \subseteq_{\mathrm{ML}} Q$ if $answer(P,T) \subseteq answer(Q,T)$ for every multi-labeled tree T.

In this paper, the central problem is the following decision problem.

- Given two conjunctive queries P and Q,
- Decide: is $P \subseteq Q$?

As pointed out in [6], the containment of k-ary queries can be PTIME reduced to the containment of Boolean conjunctive queries, i.e., queries without free variables. The same reduction works for positive XPath and for containment over multi-labeled trees. Thus, in the remainder of this paper we concentrate on Boolean query contai nment only.

Removing the attribute value comparisons. In our upper bound proofs we will treat the attribute value comparisons as ordinary labels, whose interpretation will be restricted by adding constraints. We make that precise using the translation $(\tilde{\cdot})$ which maps each $@_a \tilde{\mathrm{op}}\ c$ to a new label $p_{@_a \mathrm{op}\ c}$. This tranlation can then be homomorphically extended to the translation $(\tilde{\cdot})$ from formulas in PosXPath$^{@}$ and CQ$^{@}$ over Σ, A and D to formulas without attribute value comparisons in respectively PosXPath and CQ over the alphabet $\Sigma \cup \{p_{@_a \mathrm{op}\ c} \mid \mathrm{op} \in \{=, \neq, <, >, \leq, \geq\}, a \in A, c \in D\}$.

PosXPath *and* CQ *with safe negation*

We define an expansion of the languages PosXPath$^{@}$ and CQ$^{@}$ (UCQ$^{@}$) with a restricted form of negation. That is, we define formulas of PosXPath$^{@,\neg^s}$ as formulas of PosXPath$^{@}$ with the additional node formulas $p \wedge \neg q_1 \wedge \ldots \wedge \neg q_k$, whenever $p, q_1, \ldots, q_k$ are labels from Σ. We define $T, n \vDash p \wedge \neg q_1 \wedge \ldots \wedge \neg q_k$ iff $p \in \rho(n)$ and $q_i \notin \rho(n), 1 \leq i \leq k$.

Similarly, formulas of CQ$^{@,\neg^s}$ (UCQ$^{@,\neg^s}$) are formulas of CQ$^{@}$ (UCQ$^{@}$) expanded with the construct $p(x) \wedge \neg q_1(x) \wedge \ldots \wedge \neg q_k(x)$, where $x \in Var$ and $p, q_1, \ldots, q_k \in \Sigma$ with semantics: $T, \theta \vDash p(x) \wedge \neg q_1(x) \wedge \neg q_k(x)$ iff $p \in \rho(\theta(x))$ and $q_i \notin \rho(\theta(x))$, for every $1 \leq i \leq k$.

For a formula from CQ$^{@,\neg^s}$ its corresponding graph query is defined in the same way as in Definition 1 with the addition that nodes can have negative labels. The notion of an embedding can also be extended for CQ$^{@,\neg^s}$. The additional clause that has to be added to Definition 2 requires preservation of negated labels:

- For every $u \in V$, if $\neg p \in \rho(u)$ then $p \notin \rho(g(u))$.

By PosXPath$^{\neg^s}$, CQ$^{\neg^s}$ and UCQ$^{\neg^s}$ we denote the fragments of PosXPath$^{@,\neg^s}$, CQ$^{@,\neg^s}$ and UCQ$^{@,\neg^s}$ without attribute value comparisons respectively.

4. Containment of PosXPath$^{@}$ and CQ$^{@}$

This section contains the main result of this article. First, in Section 4.1 we show that containment for PosXPath and CQ expanded with safe negation are in coNP and Π_2^P respectively. Next we show that containment for these fragments expanded with attribute value comparisons remains the same by a polynomial reduction to the corresponding fragments without attribute value comparisons. This result holds under the assumption that attribute values come from a dense linear order without endpoints. In Section 4.3 we show that imposing different constraints on the linear domain of attribute values does not impact the complexity. However, making attributes required everywhere in a tree increases the complexity of containment, as shown in Section 4.4.

4.1. Containment of Positive Xpath and CQs with safe negation

In Section 4.2 we will reduce the containment problem for PosXPath$^{@}$ and CQ$^{@}$ to that of PosXPath$^{\neg^s}$ and UCQ$^{\neg^s}$. The next theorem shows that adding safe negation to PosXPath and UCQ does not make the containment problem harder. The argument is similar to the one in [6], but additional care needs to be taken when we deal with negation.

Theorem 1. *The containment problem over multi-labeled trees for* PosXPath$^{\neg^s}$ *and* UCQ$^{\neg^s}$ *is in* coNP *and* Π_2^P *respectively.*

Proof. In both cases the proof strategy is the same. Throughout the proof we assume that we deal with multi-labeled trees without attributes. Our goal is to show that whenever $\varphi \not\subseteq \psi$, there is a small (polynomial in φ and ψ) counterexample witnessing this fact. In the proof, we start with an arbitrary counterexample T, and shrink it in two steps: in the first step (creating T^*), we roughly restrict T to the image of φ and intermediate nodes. This can still be too large. In the second step we shrink long paths between image nodes.

Let $\varphi = \bigvee_i \varphi_i$ and $\psi = \bigvee_j \psi_j$ be UCQ$^{\neg^s}$ formulas. Let $T = (N, E, <, r, \rho)$ be a tree such that $T \vDash \varphi$ and $T \nvDash \psi$. Then there exist i and an embedding $e: Nodes(G_{\varphi_i}) \to T$, where G_{φ_i} is the query graph of φ_i. By $e(G_{\varphi_i})$ we denote the image of $Nodes(G_{\varphi_i})$. If $G_{\varphi_i} \vDash u_1 <_f u_2$, then there must exist nodes x_1 and x_2 such that $e(u_1)(E^{-1})^* x_1 <^+ x_2 E^* e(u_2)$ in T. We call such x_1 and x_2 *knee-nodes* for G.

Our aim is to create a small tree out of T which is still a counterexample. For the first "shrinking step", we color nodes that we must keep. We use three colors: $\{I, V, H\}$.

- Mark the root r with I,
- If $x \in e(G_{\varphi_i})$, mark x with I ("image" nodes),
- If $G_{\varphi_i} \vDash u_1 <_f u_2$, then there must exist knee-nodes x_1 and x_2. Mark x_1 and x_2 by I too,
- If there exist two nodes x and y marked by I such that $T \vDash xE^+y$ and there is no node z marked by I with $T \vDash xE^+z \wedge zE^+y$, then mark all the nodes on the path from x to y by V ("vertical" nodes),
- If there exist two nodes x and y marked by I or V such that $T \vDash x <^+ y$ and there is no node z marked by I or V with $T \vDash x <^+ z \wedge z <^+ y$, then mark all the sibling nodes between x and y by H ("horizontal" nodes),

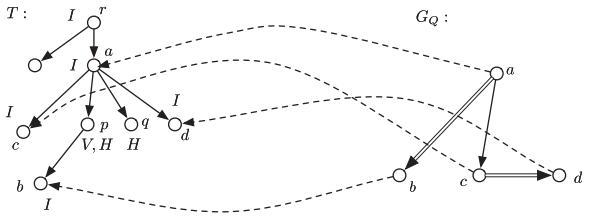

Fig. 1. The tree T and the query graph corresponding to Q from Example 1.

- Let $T^* = (N^*, E^*, <^*, r^*)$ be the substructure of T restricted to the nodes marked by I, H or V.
- Let $\rho^*(n) = \rho(n)$ for $n \in I$ and $\rho^*(n) = \varnothing$ otherwise.

Example 1. Let T be a tree as in Fig. 1 and Q the Boolean conjunctive query $\exists xyzw(a(x) \wedge Descendant(x,y) \wedge b(y) \wedge Child(x,z) \wedge c(z) \wedge NextSibling^+(z,w) \wedge d(w))$. The corresponding query graph G_Q is depicted in Fig. 1, where the downward and horizontal double line arrows denote E^+ and $<^+$ respectively, and the single line arrow denotes E. The embedding e is defined by the dashed arrows. The marking of nodes of T with the colors $\{I, V, H\}$ is depicted in Fig. 1.

Claim 1. *$T^* = (N^*, E^*, <^*, r^*, \rho^*)$ is a tree and $T^* \models \varphi$ and $T^* \not\models \psi$.*

Proof of Claim 1. It is easy to check that adding the V and H nodes to I is the minimum needed to ensure that T^* is a tree. First, we argue that $T^* \models \varphi_i$. Since we maintained the image of G_{φ_i} (nodes labeled by I), we have that e is a mapping from $Nodes(G_{\varphi_i})$ to N^*. The node labels are preserved under e since we did not change the labeling of I nodes. Let $\langle x, y\rangle$ be an edge in G_{φ_i}. If $G_{\varphi_i} \models xEy$, or $G_{\varphi_i} \models x < y$ then $e(x)$ and $e(y)$ are in respectively child or next sibling relation in T^* since both nodes are labeled with I and they were in that relation in T. If $G_{\varphi_i} \models xE^+y$ or $G_{\varphi_i} \models x <^+ y$, then $e(x)$ and $e(y)$ are in the corresponding relations in T^* since the intermediate vertical (V) and horizontal (H) nodes were kept. In case $G_{\varphi_i} \models x <_f y$ we have $T^* \models e(x) <_f e(y)$ since we kept the knee-nodes which witness the following relation in T. Thus, we obtain $T^* \models \varphi$.

Now we show $T^* \not\models \psi$. Suppose to the contrary that $T^* \models \psi$. Then there exists an embedding g of G_{ψ_j} into T^* for some j. Because T^* is a substructure of T, g is also a mapping of $Nodes(G_{\psi_j})$ into $Nodes(T)$ that preserves the edge relation. We show that g also preserves the labels. Note that by definition of T^*, the label $\rho^*(n)$ is either equal to $\rho(n)$, when $n \in I$, or empty otherwise. Positive labels are always preserved: $\rho_j(u) \subseteq \rho^*(g(u)) \subseteq \rho(g(u))$ for every node $u \in Nodes(G_{\psi_j})$, where ρ_j is the labeling function of G_{ψ_j}. We show that negative labels are preserved too. Let $u \in Nodes(G_{\psi_j})$. If $\neg p \in \rho_j(u)$, we have that $p \notin \rho^*(g(u))$, since g preserves negative labels. Since negation is safe, there must exist a label $q \in \rho_j(u)$, which implies $q \in \rho^*(g(u))$, and, thus, $\rho^*(g(u))$ is not empty. In this case $\rho^*(g(u)) = \rho(g(u))$, and thus $p \notin \rho(g(u))$ as required. Thus, $T \models \psi$ which is a contradiction. □

Next, we prove two crucial lemmas. In particular, the following lemma claims that if we have a tree T with a long enough non-branching vertical path, where each node has the empty label, and a query Q with $T \models Q$, then the path can be extended even more while preserving the fact that Q is true in the tree. We use the contrapositive of the lemma to *shrink* such long paths while keeping the query ψ *false* in the smaller tree. The same reasoning applies for horizontal paths.

Lemma 1 (*V-path*). *Let G be a query graph with labels from Σ and $T = (N, E, <, \rho, r)$ a tree such that there is an embedding of G into T. Suppose $u_1 E u_2 E \ldots E u_n$ is a path in T, such that*

- *$\rho(u_i) = \varnothing$, for every $i \in \{1, \ldots, n\}$,*
- *If $T \models u_i E x$, then $x = u_{i+1}$ for $i < n$,*
- *$n > |Nodes(G)|$.*

Let $\hat{T}$ be the tree obtained from T by inserting a node with the empty label in the middle of the path, i.e., *by making u_m the parent and u_{m+1} a child of the new node, where $n = 2m$ (when n is even) or $n = 2m - 1$ (when n is* odd). *Then there exists an embedding from G into $\hat{T}$.*

Proof. Let $G = (V, E, E^+, <, <^+, <_f, \rho)$ be the given query graph and g an embedding of G into T. Since the length n of the path is strictly greater than the number of nodes in G, there must exist an index $k \le n$ such that $u_k \notin g(V)$, $k \in \{1, \ldots, n\}$. Let $T' = (N', E', <', \rho', r')$ be the tree defined as follows:

- $r' = r$,
- $N' = N \cup \{u'_k\}, u'_k \notin N$,
- $E' = (E \setminus \{(u_k, x) \in E | x \in N\}) \cup \{(u_k, u'_k)\} \cup \{(u'_k, x) | T \models u_k E x\}$,
- $<' = <$,
- For every node $v \in N$, $\rho'(v) = \rho(v)$, and $\rho'(u'_k) = \varnothing$.

We prove that in fact the same g is an embedding of G into T' [1]. First, from the definition of T' we obtain the following properties.

Claim 2. *Let T be from the statement of Lemma 1 and T' as defined above. Then*

(i) *If $T \models xE^+y$, then $T' \models xE'^+y$,*
(ii) *If $x \neq u_k$ and $T \models xEy$, then $T' \models xE'y$,*
(iii) *If $T \models x < y$ (resp. $T \models x <^+ y$ and $T \models x <_f y$), then $T' \models x <' y$ (resp. $T' \models x <'^+ y$ and $T' \models x <'_f y$).*

Proof of Claim 2. All items except (i) are immediate by the definition of E'. For (*i*), let $T \models xE^+y$. We then consider two cases: First suppose $T \models u_k E y$. Then $T \models xE^*u_k$ and thus $T' \models xE'^*u_k$. Since $T' \models u_k E' u'_k$ and $T' \models u'_k E' y$, we have $T' \models xE'^+y$. In the case that $T \not\models u_k E y$, we obtain $T' \models xE'^+y$ by the definition of E'. □

We prove that $g: V \to N'$ is an embedding.

Preservation of labels follows from the fact that the image of g is in N and g is an embedding of G into T and thus preserves labels. We show that g still preserves the edge relations. Let xEy hold in G. Then it holds that

[1] Note that the defined T' is isomorphic to the "real" intended tree T' where the nodes are sequences of natural numbers. Here, we treat g as the old g composed with the isomorphism.

$g(x) \neq u_k$ as u_k is not in the image of g. Since g is an embedding of G into T, it holds $T \models g(x)Eg(y)$. Then by Claim 2(ii), it holds $T' \models g(x)E'g(y)$.

Let $G \models xE^+y$. Since g is an embedding of G into T, we have $T \models g(x)E^+g(y)$. By Claim 2 (i), it follows that $T' \models g(x)E'^+g(y)$. Preservation of the relations $<$, $<^+$ and $<_f$ under g follows from Claim 2 (iii).

Now let $\hat{T}$ be the tree defined in the statement of the Lemma. Formally, $\hat{T} = (\hat{N}, \hat{E}, \hat{<}, \hat{\rho}, \hat{r})$ is defined as follows.

- $\hat{r} = r$,
- $\hat{N} = N \cup \{u'_m\}, u'_m \notin N$,
- $\hat{E} = (E \setminus \{(u_m, x) \in E | x \in N\}) \cup \{(u_m, u'_m)\} \cup \{(u'_m, x) | T \models u_m E x\}$,
- $\hat{<} = <$,
- For every node $v \in N$, $\hat{\rho}(v) = \rho(v)$, and $\hat{\rho}(u'_m) = \varnothing$.

The trees $\hat{T}$ and T' are isomorphic. Recall the indexes m and k from the definitions of $\hat{T}$ and T'. We define a mapping $f: N' \to \hat{N}$ as follows.

- If $m \leq k$, then

$$f(v) = \begin{cases} v & \text{if } v \in N \setminus \{u_{m+1}, \ldots, u_k\}, \\ u'_m & \text{if } v = u_{m+1}, \\ u_{i-1} & \text{if } v = u_i, m+1 < i \leq k, \\ u_k & \text{if } v = u'_k. \end{cases}$$

- If $m > k$, then

$$f(v) = \begin{cases} v & \text{if } v \in N \setminus \{u_{k+1}, \ldots, u_m\}, \\ u_{k+1} & \text{if } v = u'_k, \\ u_{i+1} & \text{if } v = u_i, k+1 \leq i < m, \\ u'_m & \text{if } v = u_{m+1}. \end{cases}$$

The function f is onto and 1–1. We show that the V-paths in T' and $\hat{T}$ are isomorphic. We consider the case $m \leq k$, the other case is similar. Let $T' \models uE'v$ for $u, v \in \{u_1, \ldots, u_k, u'_k, u_{k+1}, u_{k+2}, \ldots, u_n\}$. We need to show that $T' \models uE'v$ iff $\hat{T} \models f(u)\hat{E}f(v)$. There are the following possible cases.

- $u = u_i$ and $v = u_{i+1}$ with $1 \leq i < m$ or $k < i < n$. In this case $f(u_j) = u_j, j \in \{i, i+1\}$. By definition of T' and $\hat{T}$ it holds that $T' \models u_i E' u_{i+1}$ and $\hat{T} \models u_i \hat{E} u_{i+1}$. Thus, $T' \models uE'v$ iff $\hat{T} \models f(u)\hat{E}f(v)$.
- $u = u_m$ and $v = u_{m+1}$. In this case $f(u_m) = u_m$ and $f(u_{m+1}) = u'_m$. By definition, it holds that $T' \models u_m E' u_{m+1}$ and $\hat{T} \models u_m \hat{E} u'_m$. Thus, $T' \models uE'v$ iff $\hat{T} \models f(u)\hat{E}f(v)$.
- $u = u_{m+1}$ and $v = u_{m+2}$. In this case $f(u_{m+1}) = u'_m$ and $f(u_{m+2}) = u_{m+1}$. By definition, it holds that $T' \models u_{m+1} E' u_{m+2}$ and $\hat{T} \models u'_m \hat{E} u_{m+1}$. Thus, $T' \models uE'v$ iff $\hat{T} \models f(u)\hat{E}f(v)$.
- $u = u_i$ and $v = u_{i+1}$ with $m+1 < i < k$. In this case, $f(u_i) = u_{i-1}$ and $f(u_{i+1}) = u_i$. By definition, it holds that $T' \models u_i E' u_{i+1}$ and $\hat{T} \models u_{i-1} \hat{E} u_i$. Thus, $T' \models uE'v$ iff $\hat{T} \models f(u)\hat{E}f(v)$.
- $u = u_k$ and $v = u'_k$. In this case, $f(u_k) = u_{k-1}$ and $f(u'_k) = u_k$. By definition, it holds that $T' \models u_k E' u'_k$ and $\hat{T} \models u_{k-1} \hat{E} u_k$. Thus, $T' \models uE'v$ iff $\hat{T} \models f(u)\hat{E}f(v)$.
- $u = u'_k$ and $v = u_{k+1}$. In this case, $f(u'_k) = u_k$ and $f(u_{k+1}) = u_{k+1}$. By definition, it holds that $T' \models u'_k E' u_{k+1}$ and $\hat{T} \models u_k \hat{E} u_{k+1}$. Thus, $T' \models uE'v$ iff $\hat{T} \models f(u)\hat{E}f(v)$.

Since $f(v) = v$ for every $v \in N \setminus \{u_1, \ldots, u_n, u'_k, u'_m\}$ and $\rho(v) = \varnothing$ for every $v \in \{u_1, \ldots, u_n, u'_k, u'_m\}$, the labels are preserved as well.

Thus, the mapping $f \circ g$ is an embedding of G into $\hat{T}$. □

Analogous to the above lemma for V-paths, we formalize one for H-paths. The crucial properties of H-paths are that their labels are empty and that all nodes in the path are leafs. We omit the proof.

Lemma 2 (H-path). *Let G be a query graph with labels from Σ and $T = (N, E, <, \rho, r)$ an ordered tree such that there is an embedding of G into T. Suppose T has a horizontal path $v_1 < v_2 < \ldots < v_n$ and v is their parent in T, where*

- $\rho(v_i) = \varnothing$ for every $i \in \{1, \ldots, n\}$,
- $V_i = \{u | T \models v_i E u\} = \varnothing$ for every $i \in \{1, \ldots, n\}$,
- $n > |Nodes(G)|$.

Let $\hat{T}$ be the tree obtained from T by inserting a node with the empty label in the middle of the horizontal path, i.e., *by making v_m the predecessor and v_{m+1} the successor of the new node, where $n = 2m$ (when n is even) or $n = 2m$-1 (when n is* odd). *Then there exists an embedding from G into $\hat{T}$.*

The proof of Theorem 1 relies on the small tree property which follows from the two lemmas above. We first show how, using Lemma 1, we can reduce the number of V-nodes. Let G_{ψ_j} be the query graph of maximal number of nodes among all G_{ψ_i}. Let $u_1 E u_2 \ldots E u_n$ be a V-path in T^* of length greater than $|Nodes(G_{\psi_j})| + 1$. Then we remove the node u_m, where $n = 2m$ (i.e., if n is even) or $n = 2m+1$ (i.e., if n is odd), from T^*, and make u_{m+1} to be the child of u_{m-1}. Let T^{**} be the resulting tree. We claim that $T^{**} \models \varphi$ and $T^{**} \not\models \psi$. The former follows from the fact that we did not change I-nodes in T^{**}. For the latter, suppose $T^{**} \models \psi$. Then there exists an embedding $g: G_{\psi_i} \to T^{**}$ for some i. Since $n - 1 > |Nodes(G_{\psi_j})| \geq |Nodes(G_{\psi_i})|$, we can apply Lemma 1 to show that there is an embedding of G_{ψ_i} to T^*, which contradicts to the fact $T^* \not\models \psi$.

Thus, we can iteratively apply the same argument to make long V-paths shorter and while preserving the fact that $T^* \not\models \psi$ and $T^* \models \varphi$. Similar for H-paths, if they are longer than $|Nodes(G_{\psi_j})| + 1$, we can apply Lemma 2 to shorten them.

Let us find out how the size of the small tree is bounded. The number of I nodes in T^* is bounded by $|Nodes(G_{\varphi_i})|$. Each I node has at most one V path above it, one H path to its right, and one H path through its children. The number of nodes in all these paths is, by the argument above, maximally $|Nodes(G_{\psi_j})| + 1$. Thus after repeated application of the Lemmas to T^* the resulting size is bounded by $O(|\varphi| \cdot |\psi|)$.

A Π_2^P algorithm for deciding the $\mathsf{UCQ}^{\neg s}$ containment then works as follows. It first guesses a tree T of size $O(|\varphi| \cdot |\psi|)$ and then checks in NP if $T \models \varphi$ and in coNP if $T \not\models \psi$. The coNP algorithm for $\mathsf{PosXPath}^{\neg s}$ works similarly. It also

guesses a tree T of polynomial size and checks if $T \models \varphi$ and $T \not\models \psi$ which can be done in PTIME [14]. □

Note that the safeness condition for negation turns out to be crucial. Indeed, in [12] it was shown that containment for tree patterns with unrestricted label negation is already PSPACE-complete.

4.2. Adding attributes

Now we are ready to provide upper bounds for our fragments with attribute value comparisons.

Theorem 2. *The containment problem over trees with attributes is*

- *in* coNP *for* $\mathsf{PosXPath}^{@,\neg^s}$,
- *in* Π_2^P *for* $\mathsf{UCQ}^{@,\neg^s}$.

Given the containment problem $\varphi \subseteq \psi$ for $\varphi, \psi \in \mathsf{PosXPath}^{@,\neg^s}$ ($\mathsf{UCQ}^{@,\neg^s}$), we reduce it to the containment problem $\varphi' \subseteq_{\mathsf{ML}} \psi'$ in $\mathsf{PosXPath}^{\neg^s}$ ($\mathsf{UCQ}^{\neg^s}$), which is known to be in coNP (Π_2^P) by Theorem 1. Thus Theorem 2 is a consequence of the following lemma.

Lemma 3. *Let* φ *and* ψ *be* $\mathsf{PosXPath}^{@,\neg^s}$ ($\mathsf{UCQ}^{@,\neg^s}$) *formulas. Then there exist* PTIME *computable* $\mathsf{PosXPath}^{\neg^s}$ ($\mathsf{UCQ}^{\neg^s}$) *formulas* φ' *and* ψ' *such that*

$$\varphi \subseteq \psi \text{ iff } \varphi' \subseteq_{\mathsf{ML}} \psi'.$$

This holds for both single-labeled and multi-labeled trees.

Proof. The idea behind the proof is as follows. We abstract away from arithmetic comparisons by replacing each of them with a new label. These labels have to obey certain constraints, like comparisons do. To this purpose, we define a list of axioms that faithfully encode these constraints.

For every $p_i, p_j \in \Sigma_p$, $p_i \neq p_j$:

$$\langle\downarrow^*\rangle(p_i \wedge p_j), \tag{Label}$$

For every $a \in \Sigma_a, c, c_1, c_2 \in \Sigma_c$, $c_1 \neq c_2$

$$\langle\downarrow^*\rangle(p_{@_a = c_1} \wedge p_{@_a = c_2}), \tag{SName}$$

$$\langle\downarrow^*\rangle(p_{@_a = c} \wedge p_{@_a \neq c}), \tag{Eq}$$

For every $a \in \Sigma_a, c \in \Sigma_c$ and R, S in $\{<, =, >\}$ with $R \neq S$,

$$\langle\downarrow^*\rangle(p_{@_a R c} \wedge p_{@_a S c}), \tag{MExcl}$$

For every $a \in \Sigma_a, c, c_1 \in \Sigma_c$ and $R \in \{\neq, \leq, \geq, <, >\}$,

$$\langle\downarrow^*\rangle(p_{@_a R c_1} \wedge \neg p_{@_a = c} \wedge \neg p_{@_a > c} \wedge \neg p_{@_a < c}), \tag{DNeg}$$

$$\langle\downarrow^*\rangle(p_{@_a \leq c} \wedge \neg p_{@_a = c} \wedge \neg p_{@_a < c}), \tag{LEQ1}$$

$$\langle\downarrow^*\rangle(p_{@_a \geq c} \wedge \neg p_{@_a = c} \wedge \neg p_{@_a > c}), \tag{GEQ1}$$

$$\langle\downarrow^*\rangle(p_{@_a = c} \wedge \neg p_{@_a \leq c}), \tag{LEQ2}$$

$$\langle\downarrow^*\rangle(p_{@_a = c} \wedge \neg p_{@_a \geq c}), \tag{GEQ2}$$

$$\langle\downarrow^*\rangle(p_{@_a < c} \wedge \neg p_{@_a \leq c}), \tag{LEQ3}$$

$$\langle\downarrow^*\rangle(p_{@_a > c} \wedge \neg p_{@_a \geq c}), \tag{GEQ3}$$

$$\langle\downarrow^*\rangle(p_{@_a < c} \wedge \neg p_{@_a \neq c}), \tag{LNEQ}$$

$$\langle\downarrow^*\rangle(p_{@_a > c} \wedge \neg p_{@_a \neq c}), \tag{GNEQ}$$

$$\langle\downarrow^*\rangle(p_{@_a \neq c} \wedge \neg p_{@_a < c} \wedge \neg p_{@_a > c}), \tag{TRI}$$

$$\langle\downarrow^*\rangle(p_{@_a \geq c} \wedge p_{@_a \leq c} \wedge \neg p_{@_a = c}), \tag{LEQGEQ}$$

For every $c_1 < c_2, c_1, c_2 \in \Sigma_c$, add the disjuncts,

$$\langle\downarrow^*\rangle(p_{@_a < c_1} \wedge \neg p_{@_a < c_2}), \tag{Order1}$$

$$\langle\downarrow^*\rangle(p_{@_a > c_2} \wedge \neg p_{@_a > c_1}), \tag{Order2}$$

$$\langle\downarrow^*\rangle(p_{@_a = c_1} \wedge \neg p_{@_a < c_2}), \tag{Order3}$$

$$\langle\downarrow^*\rangle(p_{@_a = c_2} \wedge \neg p_{@_a > c_1}). \tag{Order4}$$

In case of $\mathsf{PosXPath}^{@,\neg^s}$ we define $\varphi' := \tilde{\varphi}$ and $\psi' := \tilde{\psi} \vee Ax$, where $\tilde{(\cdot)}$ replaces comparisons with labels (definition is in Section 3) and Ax is the disjunction of the formulas mentioned above. In the definition of Ax, Σ_p, Σ_a and Σ_c are respectively the sets of labels, attributes and constants appearing in φ or ψ. We use the abbreviation $\langle\downarrow^*\rangle\theta = \theta \vee \langle\downarrow^+\rangle\theta$. Note that the formula Ax is in $\mathsf{PosXPath}^{\neg^s}$. In case of $\mathsf{UCQ}^{@,\neg^s}$ the translation $(\cdot)'$ is defined essentially the same. The only difference is that we take $\exists x.TR_x(Ax)$ instead of Ax. Notice that the resulting formulas φ' and ψ' are in $\mathsf{UCQ}^{\neg^s}$.

We first argue that the size of Ax is $O((|\varphi|+|\psi|)^3)$. Since the sets Σ_p, Σ_a and Σ_c are respectively the sets of labels, attributes and constants appearing in φ or ψ, their sizes are bounded by the combined size $|\varphi|+|\psi|$. The axioms in Ax are in fact axiom schemas. Each schema has at most 3 parameters from Σ_p, Σ_a and Σ_c and 1 parameter from the set of possible operators from $\{=, \neq, <, >, \leq, \geq\}$. Thus each schema stands for at most $6 \cdot (|\varphi|+|\psi|)^3$ disjuncts. The number of axioms and their size does not depend on φ and ψ. Whence the size of Ax is bounded by $O((|\varphi|+|\psi|)^3)$.

We give some intuition behind Ax. In order to prove this lemma, we need to show that there is a counterexample for $\varphi \subseteq \psi$ iff there is a counterexample for $\varphi' \subseteq_{\mathsf{ML}} \psi'$. Note that every counterexample tree T for $\varphi' \subseteq_{\mathsf{ML}} \psi'$ must refute every disjunct (axiom) in Ax. Intuitively, the axioms enforce the following properties of T:

- (Label): each node has at most one label from Σ_p,
- (SName): each attribute of a node can take only at most one value,
- (MExcl), (Eq): there is no inconsistent comparison,
- (DNeg): if a node contains a comparison with a constant, then it must contain a comparison with all other constants from Σ_c,
- (LEQ1)–(LEQGEQ): the natural interaction between the comparisons with a constant,
- (Order1)–(Order4): the order is preserved.

The following claim is crucial for constructing a counterexample tree for $\varphi \subseteq \psi$ from a counterexample for $\varphi' \subseteq_{\mathsf{ML}} \psi'$.

Claim 3. *Let $T=(N,E,<,r,\rho)$ be a multi-labeled tree over Σ' such that $T,r\not\models Ax$. Then for every $a\in\Sigma_a, c\in\Sigma_c$, node $n\in N$, exactly one of the following holds.*

(i) *there is no $p_{@_a \mathrm{op} c}\in\rho(n)$ for every $\mathrm{op}\in\{=,\neq,\geq,\leq,<,>\}$,*
(ii) *there is exactly one $p_{@_a = c}\in\rho(n)$ and for every $c_1\in\Sigma_c$ it holds that $p_{@_a \mathrm{op} c_1}\in\rho(n)$ iff $D\models c\,\mathrm{op}\,c_1$,*
(iii) *there is no $p_{@_a \mathrm{op} c}\in\rho(n)$ and there exists $c'\in D\setminus\Sigma_c$ such that for every $c_1\in\Sigma_c$ it holds that $p_{@_a \mathrm{op} c_1}\in\rho(n)$ iff $D\models c'\ \mathrm{op} c_1$.*

Proof of Claim. Let T be as stated in the Claim, $a\in\Sigma_a$ an attribute name, $c\in\Sigma_c$ a constant, $n\in N$ a node in T. Assume that there exists $p_{@_a \mathrm{op} c}$ in $\rho(n)$. Otherwise, item (i) holds. Assume that op is in fact "=". Because $T,r\not\models Ax$, the formula (SName) is false and thus there cannot be another $p_{@_a = c_1}$ in $\rho(n)$.

We show, for every $c_1\in\Sigma_c$ and $\mathrm{op}\in\{=,\neq,>,<,\geq,\leq\}$, that $p_{@_a \mathrm{op} c_1}\in\rho(n)$ iff $D\models c\,\mathrm{op}\,c_1$.

(1) op is $=$. Assume $p_{@_a = c_1}\in\rho(n)$, then we have $c=c_1$ by (SName). The converse implication holds since $p_{@_a = c}\in\rho(n)$ by the assumption.
(2) op is $\neq$. Assume $p_{@_a \neq c_1}\in\rho(n)$. By (Eq), it follows that $c_1\neq c$. Thus, $D\models c\neq c_1$. Conversely, assume $D\models c\neq c_1$. It means that either $c>c_1$ or $c<c_1$. First assume that $c>c_1$. Since $p_{@_a = c}\in\rho(n)$ and $c>c_1$, by (Order4) we obtain $p_{@_a > c_1}\in\rho(n)$. Then, by (GNEQ), it follows that $p_{@_a \neq c_1}\in\rho(n)$, as desired. Similarly for the case $c<c_1$, using (Order3) and (LNEQ), we can show $p_{@_a \neq c_1}\in\rho(n)$.
(3) op is $>$. Assume $p_{@_a > c_1}\in\rho(n)$. We show that $D\models c>c_1$. Suppose the opposite, i.e., either $c=c_1$ or $c_1>c$. In the first case, it would mean that both $p_{@_a = c}$ and $p_{@_a > c}$ occur in $\rho(n)$, which is a contradiction with (MExcl). In the second case, by (Order3), both $p_{@_a < c_1}$ and $p_{@_a > c_1}$ are in $\rho(n)$, which is again a contradiction with (MExcl). Now suppose $D\models c>c_1$. Then by (Order4), it follows that $p_{@_a > c_1}\in\rho(n)$, as needed.
(4) op is $<$. Similar to the previous case.
(5) op is $\geq$. Assume $p_{@_a \geq c_1}\in\rho(n)$. If $c_1=c$, then we immediately obtain $D\models c\geq c_1$. Now suppose $c\neq c_1$ and we show that $D\models c\geq c_1$. Suppose the opposite, i.e., $D\models c<c_1$. Then by (Order3), we have $p_{@_a < c_1}\in\rho(n)$. By (LEQ3), it implies that $p_{@_a \leq c_1}\in\rho(n)$, which in turn by (LEQGEQ) implies that $p_{@_a = c_1}\in\rho(n)$. The latter is a contradiction with (SName).
Now assume $D\models c\geq c_1$. This means that either $c=c_1$ or $c>c_1$ in D. In the first case, we have $p_{@_a = c}\in\rho(n)$ by the assumption. Thus by (GEQ2), we have that $p_{@_a \geq c}\in\rho(n)$. In the second case, similarly to the case when op is $>$, we can show that $p_{@_a > c_1}\in\rho(n)$. Then by (GEQ3), we have $p_{@_a \geq c_1}\in\rho(n)$, as needed.
(6) op is $\leq$. Similar to the previous case.

Thus we have proved item (ii).

Let us consider (iii). Now there is no $p_{@_a = c}\in\rho(n)$ for any $c\in D$. We define $c_1=\max\{c\mid p_{@_a > c}\in\rho(n)\}$ and $c_2=\min\{c\mid p_{@_a < c}\in\rho(n)\}$. If the former set is empty, we let $c_1=-\infty$, and if the latter set is empty, we let $c_2=+\infty$. We claim that at least one of c_1 and c_2 is finite. Indeed, there must be $p_{@_a \mathrm{R} c}$ in $\rho(n)$ for some $R\in\{\neq,<,>,\leq,\geq\}$ since otherwise item (i) would hold. By DNeg it follows that either $p_{@_a < c}$ or $p_{@_a > c}$ is in $\rho(n)$, which means that c_1 or c_2 is finite.

We also claim that $c_1<c_2$. It is trivially true if one of c_1 and c_2 is infinity, thus assume both are finite. If $c_1=c_2$, then both $p_{@_a > c_1}$ and $p_{@_a < c_1}$ appear in $\rho(n)$ at the same time, which is forbidden due to (MExcl). If $c_1>c_2$, then by (Order1), both $p_{@_a > c_1}$ and $p_{@_a < c_1}$ are in $\rho(n)$, which is again forbidden by (MExcl).

Now, since $c_1<c_2$ and the assumption that D is a dense order, there exists $c'\in D$ such that $c_1<c'<c_2$.

We claim that $c'\notin\Sigma_c$. Suppose the opposite. Then since $p_{@_a = c'}\notin\rho(n)$ by the assumption and the fact that $p_{@_a \mathrm{R} c''}\in\rho(n)$ for some R and c'' (otherwise we would be in case (i)), we obtain either $p_{@_a < c'}\in\rho(n)$ or $p_{@_a > c'}\in\rho(n)$, by (DNeg). If $p_{@_a < c'}\in\rho(n)$, then $D\models c'\geq c_2$ which contradicts to the fact that $D\models c'<c_2$. Similarly, if $p_{@_a > c'}\in\rho(n)$, then $D\models c'\leq c_1$ which contradicts with $D\models c'>c_1$.

We now show that for every $c\in\Sigma_c$ and $\mathrm{op}\in\{=,\neq,>,<,\geq,\leq\}$, $p_{@_a \mathrm{op} c}\in\rho(n)$ iff $D\models c'\mathrm{op}\ c$.

(1) op is $=$. The equivalence holds since for every $c\in\Sigma_c$, there is no $p_{@_a = c}$ in $\rho(n)$ and $D\not\models c'=c$ since $c'\notin\Sigma_c$.
(2) op is $\neq$. Assume $p_{@_a \neq c}\in\rho(n)$. Then we have that $D\models c'\neq c$ because c' is not in Σ_c. Conversely, assume $D\models c'\neq c$. By (DNeg), since there is no $p_{@_a = c}$ in $\rho(n)$ for every $c\in\Sigma_c$ and there is $p_{@_a \mathrm{R} c_1}$ for some c_1, it follows that either $p_{@_a < c}\in\rho(n)$ or $p_{@_a > c}\in\rho(n)$. Applying (LNEQ) or (GNEQ) respectively, we obtain $p_{@_a \neq c}\in\rho(n)$, as desired.
(3) op is $>$. Assume $p_{@_a > c}\in\rho(n)$. Then $D\models c'>c$ by definition of c'. Conversely, assume $D\models c'>c$. We show that $p_{@_a > c}\in\rho(n)$. Since there is no $p_{@_a = c}\in\rho(n)$ and there exists $p_{@_a \mathrm{R} c_1}\in\rho(n)$, we obtain either $p_{@_a < c}$ or $p_{@_a > c}$ in $\rho(n)$, by (DNeg). The first case is impossible since it would imply $c'<c$ which contradicts the assumption. Thus we have $p_{@_a > c}\in\rho(n)$, as desired.
(4) op is $<$. Similar to the previous case.
(5) op is $\geq$. If $p_{@_a \geq c}\in\rho(n)$, then by (GEQ1) either $p_{@_a = c}\in\rho(n)$ or $p_{@_a > c}\in\rho(n)$. The first case is impossible by the assumption. In the second case, we obtain that $D\models c'>c$ by definition of c'. Thus, $D\models c'\geq c$.
Now assume $D\models c'\geq c$. This means that either $c'=c$ or $c'>c$ in D. The first case is impossible, since $c'\notin\Sigma_c$. In the second case, as with the case $\mathrm{op}=$ ">", we can show that $p_{@_a > c}\in\rho(n)$. Then by (GEQ3), it holds that $p_{@_a \geq c}\in\rho(n)$.
(6) op is $\leq$. Similar to the previous case.

This concludes the proof of the claim. □

We now prove that $\varphi\subseteq\psi$ iff $\varphi'\subseteq_{\mathsf{ML}}\psi'$,

($\Rightarrow$) Let $T=(N,E,<,r,\rho)$ be a multi-labeled tree such that $T,r\models\varphi'$ and $T,r\not\models\psi'$. Note that then $T,r\not\models Ax$. Then we define a single-labeled tree $T':=(N,E,<,r,\rho',att)$, where att is a partial function assigning a value in D to a given node and an attribute name, as follows:

- For $p\in\Sigma_p$, $\rho'(n)=p$ iff $p\in\rho(n)$. If there is no $p\in\Sigma_p$ such that $p\in\rho(n)$, we set $\rho'(n)=z$ for a fresh symbol z.

- $att(n,a) = \begin{cases} \text{undefined} & \text{if there is no } p_{@_a \circ \mathrm{op} c_1} \text{ in } \rho(n), \\ c & \text{if } p_{@_a = c} \in \rho(n), \\ c' & \text{from Claim3, (iii), otherwise.} \end{cases}$

We claim that T' is well defined. Indeed, (Label) ensures that every node is labeled by exactly one label from Σ_p or by z. Morever, the function att is well defined since exactly one of the conditions in the definition of att is fulfilled, according to Claim 3. By induction, using Claim 3, we can show that for every θ, $T, n \models \tilde{\theta}$ iff $T', n \models \theta$. Thus, it follows $T', r \models \varphi$ and $T', r \not\models \psi$ which was desired.
($\Leftarrow$) Let $T = (N, E, <, r, \rho, att)$ be a single-labeled tree such that $T \models \varphi$ and $T \not\models \psi$. We define the tree $T' := (N, E, <, r, \rho')$, where ρ' is defined as follows:

- For $p \in \Sigma_p$, $p \in \rho'(n)$ iff $\rho(n) = p$,
- $p_{@_a = c} \in \rho'(n)$ iff $att(n,a) = c$,
- $p_{@_a \circ \mathrm{op} c} \in \rho'(n)$ iff $D \models att(n,a) \circ \mathrm{op} c$ for $\mathrm{op} \in \{\neq, \leq, \geq, <, >\}$, $c \in \Sigma_c$.

It is straightforward to check that T' does not satisfy any of the disjuncts in Ax. Thus, we obtain $T' \models \varphi'$ and $T' \not\models \psi'$.□

The same argument goes through for multi-labeled trees, except that we must not include formulas *(Label)* juncts of Ax.

4.3. Restricting the attribute domain

We now show the same complexity results for various restrictions on the domain of attribute values. A linear order D has a smallest (largest) element if there exists $c' \in D$ such that $c' \leq c$ ($c \geq c'$) for every $c \in D$. We say D has an endpoint if there exists a smallest or a largest element in D. D is discrete if any point which has a successor also has an immediate successor. D is dense linear order if for every $x < y$ in D there exists $z \in D$ such that $x < z < y$.

Proposition 2. *Let D be one of the following linear* orders:

(i) *finite*,
(ii) *discrete*,
(iii) *dense or discrete with one or two endpoints.*

Then the containment problem for $\mathsf{PosXPath}^{@,\neg^s}$ *and* $\mathsf{UCQ}^{@,\neg^s}$ *over single-labeled trees with the domain of attribute values D is in* coNP *and Π_2^p respectively.*

Proof. Let φ and ψ be $\mathsf{PosXPath}^{@,\neg^s}$ ($\mathsf{UCQ}^{@,\neg^s}$) formulas over D and $\Sigma_c \subseteq D$, Σ_a and Σ_p the sets of constants, attribute names and labels in Σ appearing in φ or ψ. We then construct in PTIME formulas φ' and ψ' over $\Sigma' = \Sigma_p \cup \{p_{@_a \circ \mathrm{op} c} \mid a \in \Sigma_a, c \in \Sigma_c, \mathrm{op} \in \{=, \neq, <, >, \leq, \geq\}\}$ such that

$$\varphi \subseteq \psi \text{ if and only if } \varphi' \subseteq_{\mathsf{ML}} \psi'. \tag{1}$$

Namely, we take $\varphi' := \tilde{\varphi}$ and $\psi' := \tilde{\psi} \vee Ax \vee Ax_k$, where $\tilde{(\cdot)}$ is defined in Section 3, Ax is from the proof of Lemma 3 and $Ax_k, k \in \{(Fin), (Discr), (End)\}$ is constructed according to the cases (i), (ii) and (iii) of the Proposition. Note that the formulas φ' and ψ' in all the cases are in fact $\mathsf{PosXPath}^{\neg^s}$ formulas. In case of $\mathsf{UCQ}^{@,\neg^s}$, the translation $(\cdot)'$ is defined essentially the same. The difference is that we use $\exists x.TR_x(Ax)$ and $\exists y.TR_y(Ax_k)$ instead. Note that the result of $(\cdot)'$ is a union of $\mathsf{CQ}^{\neg^s}$ formulas. The upper bounds then follow from Theorem 1.

Now we construct the formulas $Ax_k, k \in \{(Fin), (Discr), (End)\}$.

(i). Assume $D = \{c_1 < c_2 < \ldots < c_k\}$ is a finite linear order. We then write down the formulas of $Ax_{(Fin)}$. It is the disjunction of the following formulas. For every $a \in \Sigma_a$, $c \in \Sigma_c$ and $\mathrm{op} \in \{=, \neq, <, >, \leq, \geq\}$:

$$\langle\downarrow^*\rangle(p_{@_a \circ \mathrm{op} c} \wedge \neg p_{@_a = c_1} \wedge \ldots \wedge \neg p_{@_a = c_k}). \tag{Fin}$$

This axiom enforces that whenever an attribute is defined, its value equals one of $c_i, 1 \leq i \leq k$. The following claim, which is easy to verify using (Fin), is crucial.

Claim 4. *Let $T = (N, E, <, r, \rho)$ be a multi-labeled tree over Σ' such that $T, r \not\models Ax \vee Ax_{(Fin)}$. Then for every $a \in \Sigma_a$, $c \in \Sigma_c$, node $n \in N$ and $\mathrm{op} \in \{=, \neq, \geq, \leq, <, >\}$, exactly one of the following holds:*

(i) *there is no $p_{@_a = c} \in \rho(n)$,*
(ii) *there is exactly one $p_{@_a = c} \in \rho(n)$ and for every $c_1 \in \Sigma_c$ it holds that $p_{@_a \circ \mathrm{op} c_1} \in \rho(n)$ iff $D \models c \circ \mathrm{op} c_1$.*

Now we prove the equivalence (1). For the direction from left to right, given a multi-labeled tree T over Σ' such that $T \models \varphi'$ and $T \not\models \psi'$, we construct a single-labeled tree with attributes T' as it was done in Lemma 3. The only difference is in the definition of the attribute function att. In our case we take

$$att(n,a) = \begin{cases} \text{undefined} & \text{if there is no } p_{@_a \circ \mathrm{op} c} \text{ in } \rho(n), \\ & \mathrm{op} \in \{=, \neq, <, >, \geq, \leq\}, c \in \Sigma_c \\ c & \text{if } p_{@_a = c} \in \rho(n). \end{cases}$$

Using Claim 4 we can show that for every $n \in T$, θ over Σ, A and D, $T, n \models \tilde{\theta}$ iff $T', n \models \theta$.

The direction from right to left of (1) can be proved exactly as in Lemma 3.

(ii). D is a discrete linear order. We assume that D is infinite, as the finite case is covered by the case (i). We take $Ax_{(Discr)}$ as the disjunction of the following formulas. For every $a \in \Sigma, c_1, c_2 \in \Sigma_c$ such that $c_1 < c_2$ in D and there is no c' in D with $c_1 < c' < c_2$,

$$\langle\downarrow^*\rangle(p_{@_a > c_1} \wedge p_{@_a < c_2}). \tag{Discr}$$

This axiom enforces the requirement that a value for a-attribute cannot be between an element in D and its immediate successor.

Similarly to Lemma 3 we can show that the reduction is correct. To this purpose we need the claim which is the exact reformulation of Claim 3 where instead of Ax we take $Ax \vee Ax_{(Discr)}$ and D is the discrete linear order.

We highlight the difference with the proof of Claim 3. The only nontrivial difference is item (iii). Assume the conditions (i) and (ii) of Claim 3 do not hold for $a \in A$ and $n \in T$. We define $c_1 = \max\{c \mid p_{@_a > c} \in \rho(n)\}$ and $c_2 = \min\{c \mid p_{@_a < c} \in \rho(n)\}$. As in Claim 3 we can show that $D \models c_1 < c_2$. Having that, there exists c' such that $c_1 < c' < c_2$. Indeed, suppose the opposite. Then both

$p_{@_a > c_1}$ and $p_{@_a < c_2}$ are in $\rho(n)$ and c_2 is the immediate successor of c_1 in D, which is a contradiction with (Discr). It follows from (DNeg) that $c' \notin \Sigma_c$. Moreover, for every $c'' \in \Sigma_c$, $p_{@_a c' \circ c''} \in \rho(n)$ iff $D \models c' \circ c''$. This can be verified as it was done in Claim 3. Thus we have proved the claim. Having this claim at hand, we can prove the equivalence (1) in the same way as in Lemma 3.

(iii). D is dense or discrete linear order with one or two endpoints. If D is dense, take $Ax_{(End)}$ as the disjunction ofthe following formulas:

If D has the least endpoint c_l, for every $a \in \Sigma_a$:

$$\langle\downarrow^*\rangle p_{@_a < c_1}. \tag{LEnd}$$

If D has the greatest endpoint c_g, for every $a \in \Sigma_a$:

$$\langle\downarrow^*\rangle p_{@_a > c_g}. \tag{REnd}$$

In case D is discrete linear order, $Ax_{(End)}$ additionally has (Discr) as a disjunct.

The axioms (LEnd) and (REnd) enforce the requirement that attributes cannot take their values outside of the bounds in D. As in the previous case, we can prove the variant of Claim 3, where we consider $Ax_{(End)}$ and D a dense or discrete linear order with one or two endpoints. We do not spell out the proof, but the crucial difference is that in item (iii) axioms (LEnd) and (REnd) ensure the fact that c' is chosen within the interval $[c_l, c_g]$.

Having this claim, we can prove the equivalence (1) in the same way as in Lemma 3.

Clearly, the constructed φ' and ψ' are PTIME computable from φ and ψ.

4.4. Lower bounds

In this section we show a number of lower bounds on containment for $\mathsf{CQ}^{@}$ and $\mathsf{PosXPath}^{@}$. The following lower bound was shown in [6].

Proposition 3 ([6]). *Containment is Π_2^p-hard for* CQ(*Child*, *Descendant*), i.e., *conjunctive queries that use only the predicates Child and Descendant.*

For $\mathsf{PosXPath}^{@}$, the coNP lower bound for containment follows from hardness of containment for tree patterns [19], which is a fragment of $\mathsf{PosXPath}^{@}$. In order to compare our results to those in [19], we follow their notation. Let $\mathsf{XP}^{\{[\,],*,//\}}$ denote the fragment of PosXPath without union and disjunction, only the $\downarrow$ step, and no occurence of the following and preceding axes. These are called *tree patterns* in the literature. Let $\mathsf{XP}^{\{[\,],//\}}$ denote $\mathsf{XP}^{\{[\,],*,//\}}$ in which no wildcard (denoted by $\top$ in PosXPath) occurs.

Containment of $\mathsf{XP}^{\{[\,],//\}}$ and $\mathsf{XP}^{\{[\,],*,//\}}$ patterns is in PTIME and coNP-complete, respectively. Let $\mathsf{XP}^{\{[\,],*,//\}}_{=,\neq}$ and $\mathsf{XP}^{\{[\,],//\}}_{=,\neq}$ denote the expansions of $\mathsf{XP}^{\{[\,],*,//\}}$ and $\mathsf{XP}^{\{[\,],//\}}$ with equality and inequality attribute value comparisons, respectively. We show that containment of $\mathsf{XP}^{\{[\,],//\}}_{=,\neq}$ patterns becomes coNP hard. Containment of $\mathsf{XP}^{\{[\,],*,//\}}_{=,\neq}$ patterns becomes PSPACE hard when interpreted over trees with at least one required attribute.

The following property is used in our lower bound arguments. The proof can be found in [19, Lemma 3].

Proposition 4. *Let L be $\mathsf{XP}^{\{[],*,//\}}$ or $\mathsf{XP}^{\{[\,],//\}}_{=,\neq}$. Let φ be an L formula and Δ a finite set of L formulas. Then there are* PTIME *computable L formulas φ' and ψ' such that*

$$\varphi \subseteq \bigvee\Delta \text{ iff } \varphi' \subseteq \psi'.$$

The same holds for the case of multi-labeled trees.

Proposition 5. *The containment problem for $\mathsf{XP}^{\{[\,],//\}}_{=,\neq}$ is* coNP*-hard.*

Proof. We reduce the 3SAT problem to the non-containment problem in $\mathsf{XP}^{\{[\,],//\}}_{=,\neq}$.

Firstly, we can use disjunction of tree patterns on the right side of the containment problem, due to Proposition 4.

Let Q be the conjunction of clauses $C_i = (X_1^i \vee X_2^i \vee X_3^i), 1 \le i \le k$ over the variables $\{x_1, \ldots, x_n\}$, where X_j^i are literals. From Q, we construct in PTIME two formulas over the signature $\Sigma = \{r, b\}$, attribute names $A = \{a_1, \ldots, a_n\}$ and an attribute domain D containing values $\{0, 1, 2\}$ as follows.

We define $\varphi := r \wedge \langle\downarrow\rangle(b \wedge @_{a_1} \neq 2 \wedge \ldots \wedge @_{a_n} \neq 2)$[2] and $\psi := \bigvee_{i=1}^{k} \langle\downarrow\rangle(b \wedge B_1^i \wedge B_2^i \wedge B_3^i)$, where $B_j^i = (@_{a_l} = 0)$ iff $X_j^i = x_l$ in C_i and $B_j^i = (@_{a_l} \neq 0)$ iff $X_j^i = \neg x_l$ in C_i.

We claim that Q is satisfiable if and only if $\varphi \not\subseteq \psi$. First assume that Q is satisfiable, i.e., there is a variable assignment $V: \{x_1, \ldots, x_n\} \to \{0, 1\}$ such that $V \models Q$. We then define the following tree $T = (\{v_1, v_2\}, \{(v_1, v_2)\}, v_1, \rho, att)$, where the labeling ρ is defined as $\rho(v_1) = \{r\}$, $\rho(v_2) = \{b\}$ and $att(v_2, a_l) = 1$ iff $V(x_l) = 1$ and $att(v_2, a_l) = 0$ iff $V(x_l) = 0$ for every $l, 1 \le l \le n$. Clearly, T satisfies φ. Suppose $T, v_1 \models \psi$. This means there exists an index i such that $T, v_1 \models \langle\downarrow\rangle(b \wedge B_1^i \wedge B_2^i \wedge B_3^i)$, which implies $T, v_2 \models B_j^i, j = 1, 2, 3$. Hence, by the definition of the attribute function we obtain that if $B_j^i = (@_{a_l} = 0)$, then $V(X_j^i) = V(x_l) = 0$ and, similarly, if $B_j^i = (@_{a_l} \neq 0)$, then $V(X_j^i) = V(\neg x_l) = 0$. Thus, we obtain $V \not\models C_i$, which is a contradiction. Thus, $T \not\models \psi$.

We now prove the converse. Assume there is a tree T with $T \models \varphi$ and $T \not\models \psi$. The former implies that there exists a child of the root of T, v such that $T, v \models b$ and the attributes $a_1, \ldots, a_n$ are defined at v. Moreover since $T \not\models \psi$, for every $i, 1 \le i \le n$ it holds that $T, v \not\models b \wedge B_1^i \wedge B_2^i \wedge B_3^i$. We define the variable assignment $V: \{x_1, \ldots, x_n\} \to \{0, 1\}$ as follows: $V(x_l) := 0$ iff $att(v, a_l) = 0$ and $V(x_l) := 1$ iff $att(v, a_l) \neq 0$. We claim that $V \models Q$. Assume the opposite, i.e., there exists a clause C_j which is mapped to 0 under V. By definition of V, it follows that $T, v \models B_1^j \wedge B_2^j \wedge B_3^j$ and therefore, $T, v \models b \wedge B_1^j \wedge B_2^j \wedge B_3^j$ which is a contradiction.

4.4.1. Required attributes

In Section 4.2 we dealt with the case when attributes are optional. We now consider the case when some attributes are required. We say that an attribute $a \in A$ is required in a tree T with domain N if the function $att: N \times \{a\} \to D$ is total. We show that when at least one attribute is required, containment of tree patterns with equality and inequality comparisons rises to PSPACE.

[2] The purpose of the inequalities $@_{a_i} \neq 2$ is to guarantee that the attribute a_i is defined in the b-node of a model of φ. We could express the same with the comparison $@_{a_i} \le 1$ or $@_{a_i} \ge 0$.

Theorem 3. *The containment problem for* $\mathsf{XP}^{\{[\,],*,//\}}_{=,\neq}$ *interpreted over trees with at least one required attribute is* PSPACE*-complete.*

Proof. We show the upper bound for $\mathsf{XP}^{\{[\,],*,//\}}_{=,\neq}$ expanded with the other equality operators (i.e., $<$, $>$, $\leq$ and $\geq$). For that, we reduce the containment problem in this fragment to containment for unions of $\mathsf{XP}^{\{[],*,//,\neg\}}$ (tree pattern formulas with unrestricted label negation) similar to Lemma 3. The additional axiom in Ax is $\langle\downarrow^*\rangle(\neg p_{@_{a=c}} \wedge \neg p_{@_{a\neq c}})$ for every required $a \in A$, where c is a constant (note that this axiom contains *unsafe* negation). This axiom enforces that the attribute a is defined everywhere in the tree. In [12] it is shown that containment for unions of $\mathsf{XP}^{\{[],*,//,\neg\}}$ is solvable in PSPACE.

For proving the lower bound we encode the corridor tiling problem, which is known to be hard for PSPACE [8]. Our lower bound proof uses the construction from the PSPACE-hardness proof for the containment problem in tree patterns with disjunction over a finite alphabet in [20].

The corridor tiling problem is formalized as follows. Let $\mathsf{Til} = (D, H, V, \overline{b}, \overline{t}, n)$ be a tiling system, where $D = \{d_1, \ldots, d_m\}$ is a finite set of tiles, $H, V \subseteq D^2$ are horizontal and vertical constraints, n is a natural number in unary notation, $\overline{b}$ and $\overline{t}$ are tuples over D of length n. Given such a tiling system, the goal is to construct a tiling of the corridor of width n using the tiles from D so that the constraints H and V are satisfied. Moreover, the bottom and the top row must be tiled by $\overline{b}$ and $\overline{t}$ respectively.

Let $a \in A$ be a required attribute. Now we construct two $\mathsf{XP}^{\{[],*,//\}}_{=,\neq}$ expressions φ and ψ such that $\varphi \not\subseteq \psi$ over trees with a required attribute a iff there exists a tiling for Til. To this purpose, we use a string representation of a tiling. Each row of the considered tiling is represented by the tiles it consists of. If the tiling of a corridor of width n has k rows, it is represented by its rows separated by the special symbol $\sharp$. Thus, a tiling is a word of the form $u_1\sharp u_2\sharp\cdots\sharp u_k\$$, where each u_i is the word of length n corresponding to the i-th row in the tiling, and $\$$ denotes the end of tiling. Note $u_1 = \overline{b}$ and $u_k = \overline{t}$.

For the sake of readability, for expression r, we use the abbreviation r^i to denote the path formula $?r;\downarrow;?r;\ldots;\downarrow;?r$ with i occurrences of r.

We then define the formulas over attributes $\{a\}$ and attribute domain containing $D \cup \{\sharp\}$.

Define φ' as

$$\langle ?(@_a = b_1);\downarrow;?(@_a = b_2);\ldots;\downarrow;?(@_a = b_n);\downarrow;?(@_a = \sharp);\downarrow^+ :$$
$$?(@_a = t_1),\downarrow;\ldots\downarrow;?(@_a = t_n);\downarrow\rangle\$.$$

Intuitively, this expression enforces a tiling to start with a path starting with $\overline{b}$ and finishing with $\overline{t}$. Now the formula ψ' defines all incorrect tilings and additional constraints. It is the disjunction of the following $\mathsf{XP}^{\{[],*,//\}}_{=,\neq}$ formulas.

(1) Incorrect length of a row.
  (1a) $\bigvee_{i=0}^{n-1}\langle\downarrow^+;?(@_a = \sharp);\downarrow;\top^i;\downarrow\rangle(@_a = \sharp)$, a row is too short,
  (1b) $\langle\downarrow^+;(@_a \neq \sharp)^{n+1}\rangle\top$, a row is too long.
(2) $\langle\downarrow^+;?(@_a \neq d_1 \wedge \ldots \wedge @_a \neq d_m \wedge @_a \neq \sharp);\downarrow^+\rangle\$$, neither the delimiter or a tile on a position,
(3) Horizontal or vertical constraints are violated.
  (3a) $\bigvee_{(d_1,d_2)\notin H}\langle\downarrow^+;?(@_a = d_1);\downarrow;?(@_a = d_2)\rangle\top$, a horizontal constraint is violated,
  (3b) $\bigvee_{(d_1,d_2)\notin V}\langle\downarrow^+;?(@_a = d_1);\downarrow;\top^n;\downarrow;?(@_a = d_2)\rangle\top$, a vertical constraint is violated.

We show that there exists a tree with a required attribute a such that $T \models \varphi'$ and $T \not\models \psi'$ iff there exists a tiling for Til.

($\Leftarrow$). Assume that there exists a tiling of the corridor. Let s be the string representation of it. Then, $s = u_1\sharp u_2\sharp\ldots\sharp u_k\$$, where $|u_i| = n, u_i \in D^n, u_1 = \overline{b}$, and $u_k = \overline{t}$. Moreover, on the one hand if $x \cdot y$, is an infix of some u_i, then $(x, y) \in H$, and on the other hand for every infix $x \cdot u' \cdot y$ of length $n+1$ of $u_i\sharp \cdot u_{i+1}$, it holds that $(x, y) \in V$. Let T_s be the corresponding tree, i.e., a single path of $|s|$ nodes $\{v_1, \ldots, v_{|s|}\}$ where the label of each node $v_i, i < |s|$ is z, the label of $v_{|s|}$ is $\$$ and attribute function is defined according to s, i.e., $att(v_i, a) = s_i$. Clearly, T_s is a model of φ' and not of ψ'.

($\Rightarrow$). Let T be a tree such that $T, r \models \varphi'$, $T, r \not\models \psi'$ and $att(n, a)$ is defined for every $n \in Nodes(T)$. Since $T, r \models \varphi'$, there must exist a path $r = v_1, \ldots, v_m$ in T such that $att(v_i, a) = b_i, 1 \leq i \leq n$ and $att(v_{m-n+i}, a) = t_j, 1 \leq j \leq n$. Moreover, either $\sharp$ or a symbol from D is in the attribute of every node $v_i, 1 \leq i < m$, according to (2).

We define a tiling function $g\colon \{0, \ldots, n-1\} \times \mathbb{N} \to D$ assigning a tile to every position in the corridor as follows: $g(i, j) = att(v_{(n+1)\times j+i+1}, a), 1 \leq i \leq n$. Indeed, this function is well defined, as (1) ensures the correct counting.

By formulas (3a) and (3b) the tiling defined by g satisfies the horizontal and vertical constraints.

We then apply Proposition 4 to remove the outermost disjunction in ψ' to obtain the equivalent containment problem $\varphi \subseteq \psi$ in $\mathsf{XP}^{\{[\,],*,//\}}_{=,\neq}$ □.

Theorem 3 provides a lower bound for the containment problem for $\mathsf{PosXPath}^{@,\neg^s}$ and $\mathsf{UCQ}^{@,\neg^s}$ over trees with required attributes. Only for tree patterns we know that the problem is PSPACE-complete. Using the same reduction as in the proof of the upper bound in Theorem 3, and the results on containment for XPath [18] and XPath with path intersection [24], we obtain EXPTIME and 2EXPTIME upper bounds for containment for $\mathsf{PosXPath}^{@,\neg^s}$ and $\mathsf{UCQ}^{@,\neg^s}$ over trees with required attributes, respectively.

However, if we restrict attributes to be required at nodes labeled with a certain symbol, then the containment is still in coNP and Π^p_2. Let $p \in \Sigma$ be a label and $a \in A$ an attribute name. We say that a is *required at label element* p if $att(n, a)$ is defined whenever $p \in \rho(n)$ for every tree T and node $n \in Nodes(T)$.

Proposition 6. *The containment problem for* $\mathsf{PosXPath}^{@,\neg^s}$ *and* $\mathsf{UCQ}^{@,\neg^s}$ *with required attributes at certain labeled nodes is in* coNP *and* Π^p_2 *respectively.*

Proof. As before, we can prove a variant of Lemma 3. Let c be a constant name. Whenever attribute a is required at nodes labelled by p we add the axiom $\langle\downarrow^*\rangle(p \wedge \neg p_{@_{a=c}} \wedge \neg p_{@_{a\neq c}})$ to the set Ax. Note that the negation is safe. This axiom is obviously sound, and it enforces that whenever p holds, at least one $p_{@_a \circ p c}$ label holds as well.

This ensures that in the construction of the tree with attribues a is defined at each p node. □

4.5. Tractable cases

In this section we consider fragments of PosXPath$^{@}$ where the containment problem remains in PTIME. It is known that containment in $\mathsf{XP}^{\{[],//\}}$ and $\mathsf{XP}^{\{[],*\}}$ is decidable in PTIME, [3,19].

Proposition 7. *Let XP^{X} be any fragment whose containment problem over multiple-labeled trees is in* PTIME. *Then the containment problem in $\mathsf{XP}_{=}^{@,X}$ over multi-labeled trees with attributes is also in* PTIME.

Proof. Let φ and ψ be formulas in $\mathsf{XP}_{=}^{@,X}$.

Our algorithm first checks (in PTIME) if φ is consistent, i.e., if it contains both $@_a = c$ and $@_a = d$ in the label of a node in φ for some $a \in A, c, d \in D$. If φ is inconsistent, we output $\varphi \subseteq \psi$. Otherwise, we proceed as in the proof of Lemma 3 by reduction to a containment of attribute-free formulas using the translation $\tilde{(\cdot)}$ and the formula (Label) only.□

5. Conclusion

We considered the containment problem for positive XPath and conjunctive queries over trees expanded with attribute value comparisons. We showed that in general attribute value comparisons do not increase the complexity of containment. The main idea behind the upper bound was to extend the small counterexample technique to positive XPath and conjunctive queries expanded with a restricted form of negation. Then by axiomatizing the needed constraints in the corresponding expanded fragment, we could abstract away the attribute value comparisons.

The complexity, however, does increase from PTIME to coNP for the fragment $\mathsf{XP}^{\{//,[\ ]\}}$ of XPath which uses child, descendant and filter expressions when we add equality and inequality comparisons. Another parameter which affects the complexity is optionality of attributes. If we restrict our trees to have at least one required attribute in every node, then the complexity rises to PSPACE. If, however, attributes are required at elements with specific labels only, the complexity of containment remains the same: coNP for positive XPath and Π_2^p for conjunctive queries.

We end with listing some open problems. Proposition 7 shows that adding equality comparison only does not affect the PTIME complexity of containment for fragments $\mathsf{XP}^{\{//,[\ ]\}}$ and $\mathsf{XP}^{\{[\ ],*\}}$. We do not know what happens when only inequality comparison is added.

For conjunctive queries over trees, it is known that the fragments CQ(*Child*) and CQ(*NextSibling*) have PTIME containment. It is open whether the complexity increases if we add attribute value comparisons.

Acknowledgements

We thank the anonymous referees for their helpful comments.

This research was supported by the Netherlands Organization for Scientific Research (NWO) under project number 612.001.012 (DEX).

References

[1] F.N. Afrati, S. Cohen, G.M. Kuper, On the complexity of tree pattern containment with arithmetic comparisons, Inf. Process. Lett. 111 (15) (2011) 754–760.
[2] F.N. Afrati, C. Li, P. Mitra, On containment of conjunctive queries with arithmetic comparisons, in: Proceedings of the 9th International Conference on Extending Database Technology Advances in Database Technology - EDBT 2004, Heraklion, Crete, Greece, March 14-18, 2004, pp. 459–476.
[3] S. Amer-Yahia, S. Cho, L. Lakshmanan, D. Srivastava, Tree pattern query minimization, VLDB J. 11 (2002) 315–331.
[4] M. Benedikt, W. Fan, F. Geerts, XPath satisfiability in the presence of DTDs, J. ACM, 55(2), 2008.
[5] H. Björklund, W. Martens, T. Schwentick, Optimizing conjunctive queries over trees using schema information, MFCS (2008) 132–143.
[6] H. Björklund, W. Martens, T. Schwentick, Conjunctive query containment over trees, J. Comput. Syst. Sci. 77 (3) (2011) 450–472.
[7] A.K. Chandra and P.M. Merlin, Optimal implementation of conjunctive queries in relational data bases, in: Proceedings of the 9th Annual ACM Symposium on Theory of Computing, May 4-6, 1977, Boulder, Colorado, USA, 1977; pp. 77–90.
[8] B.S. Chlebus, Domino-tiling games, J. Comput. Syst. Sci. 32 (3) (1986) 374–392.
[9] C. David, A. Gheerbrant, L. Libkin, W. Martens, Containment of pattern-based queries over data trees, ICDT, 2013.
[10] A. Deutsch, V. Tannen, Containment and integrity constraints for XPath, in: M. Lenzerini, D. Nardi, W. Nutt, D. Suciu (Eds.), Proceedings of CEUR Workshop, KRDB, vol. 45, 2001, ⟨CEUR-WS.org⟩.
[11] A. Deutsch, V. Tannen, XML queries and constraints, containment and reformulation, Theor. Comput. Sci. 336 (1) (2005) 57–87.
[12] A. Facchini, Y. Hirai, M. Marx, E. Sherkhonov, Containment for conditional tree patterns, Logical Methods in Computer Science 11 (2) 2015.
[13] C. Farré, W. Nutt, E. Teniente, T. Urpí, Containment of conjunctive queries over databases with null values, in: Proceedings of the 11th International Conference, Database Theory - ICDT 2007, Barcelona, Spain, January 10-12, 2007, pp. 389–403.
[14] G. Gottlob, C. Koch, R. Pichler, Efficient algorithms for processing XPath queries, ACM Trans. Database Syst. 30 (2) (2005) 444–491.
[15] G. Gottlob, C. Koch, K.U. Schulz, Conjunctive queries over trees, J. ACM, 53 , 2006, (2):238–272.
[16] J. Hidders, Satisfiability of XPath expressions, in: G. Lausen, D. Suciu (Eds.), DBPL, of Lecture Notes in Computer Science, vol. 2921, Springer, 2003, pp. 21–36.
[17] A.C. Klug, On conjunctive queries containing inequalities, J. ACM, 35, 1988, (1):146–160.
[18] M. Marx, Conditional Xpath, ACM Trans. Database Syst. 30 (4) (2005) 929–959.
[19] G. Miklau, D. Suciu, Containment and equivalence for a fragment of Xpath, J. ACM, 51 , 2004, (1):2–45.
[20] F. Neven, T. Schwentick, On the complexity of Xpath containment in the presence of disjunction, DTDs, and variables, Log. Methods Comput. Sci. 2 (3) 2006.
[21] W. Nutt, Ontology and database systems: foundations of database systems, 2013. Teaching material. ⟨http://www.inf.unibz.it/∼nutt/Teaching/ODBS1314/ODBSSlides/3-conjQueries.pdf⟩.
[22] D. Olteanu, H. Meuss, T. Furche, F. Bry, Xpath:looking forward, EDBT, 2002, pp. 109–127.
[23] E. Sherkhonov M. Marx, Containment for tree patterns with attribute value comparisons, WebDB, 2013.
[24] B. ten Cate, C. Lutz, The complexity of query containment in expressive fragments of XPath 2.0, J. ACM, 56(6), 2009.
[25] J.D. Ullman, Information integration using logical views, Theor. Comput. Sci. 239 (2) (2000) 189–210.
[26] R. van der Meyden, The complexity of querying indefinite data about linearly ordered domains, J. Comput. Syst. Sci. 54 (1) (1997) 113–135.
[27] F. Wei, G. Lausen, Containment of conjunctive queries with safe negation, ICDT 2003, 2003, pages 343–357.
[28] P.T. Wood, Containment for XPath fragments under DTD constraints, in: ICDT, 2003, pp. 297–311.