Containment for tree patterns with attribute value comparisons

Evgeny Sherkhonov
ISLA, University of Amsterdam
e.sherkhonov@uva.nl

Maarten Marx
ISLA, University of Amsterdam
maartenmarx@uva.nl

ABSTRACT

Tree patterns (TP) is a simple and widely used fragment of XPath. The problem of containment in TP has been extensively studied previously. It was shown that the containment problem ranges from PTime to PSpace depending on the available constructs.

In this paper we study the complexity of the containment problem for tree patterns with attribute value comparisons. We show that the complexity ranges between PTime and PSpace. We distinguish the parameters which have to be taken into account in the containment problem: (i) available axes, (ii) type of comparisons (e.g. $\neq$-comparions), (iii) the underlying domain for attribute values (e.g. linear dense order) and (iv) optionality of attributes.

Categories and Subject Descriptors

H.2.3 [**Database Management**]: Languages

General Terms

Languages, Theory

Keywords

XML, Tree Patterns, Containment

1. INTRODUCTION

Tree patterns (TP) is a natural query language for XML which is used in many XML data management problems. They can be seen as the conjunctive downward fragment of XPath. Equivalently, they can be seen as trees, see Figure 1. The tree pattern containment and equivalence problems are essential in the context of query optimization. In [2] it was shown that the containment problem for basic tree patterns (that is, tree patterns constructed using child, descendent and filter expression) is solvable in PTime. Adding the wildcard rises the complexity to coNP [12]. Assuming a finite alphabet further lifts the complexity to PSpace [13].

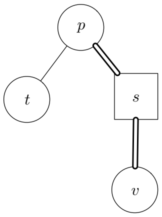

Figure 1: The tree pattern corresponding to the XPath expression `/p[t]//s[.//v]`. The square box is the output node.

In this paper we look at tree patterns expanded with attribute value comparisons. In [1], the containment problem with such comparison has been studied and it was shown that the containment problem is Π_2^P-complete. However, the Π_2^P-hardness proof heavily uses comparisons between attribute values of different nodes, a feature which 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 semi-interval attribute constraints. For an attribute a and constant c, an attribute constraint $@_a$ $\texttt{op}$ c is left semi-interval if $\texttt{op} \in \{<, \leq, =\}$.

The main result of this paper is that the containment problem of tree patterns expanded with *both* left and right semi-interval constraints is also in coNP. Furthermore, this upper bound holds for the cases when we make certain assumption on the underlying domain D for attribute values. More precisely, we show that all the complexity results still hold for the cases when D is dense or discrete infinite linear order, with or without endpoints, or finite linear order. As another parameter for the containment problem, we consider optionality of attributes: we show that the complexity rises to PSpace when at least one attribute is required in every node. If constraints stating that attributes are required at nodes with a certain label (as can be expressed in DTD's) are added, containment remains in coNP. In all our lower bound proofs, we only use the operators $=$ and $\neq$.

All the coNP upper bounds are obtained from a suitable polynomial reduction to the containment problem in $\mathsf{TP}^{\neg^g}$ (tree patterns with guarded label negation) over multi-labeled trees. Guarded label negation is the construct $p \setminus \{q_1, \ldots, q_k\}$ meaning a p-node which does not contain labels $q_1, \ldots, q_k$.

Table 1 summarizes our main results. The paper is organized as follows. Section 2 contains preliminaries, Sec-

	lower bounds	**upper bounds**
operators	$=, \neq$	all
attribute-free	PTime [2]	coNP [12]
optional attributes	coNP, (Prop 3)	coNP, (Prop 2)
required attributes	PSpace, (Prop 3)	PSpace

Table 1: Complexity results for tree patterns with attribute value comparisons.

tion 3 all results mentioned in Table 1 and Section 4 discusses tractable cases and tractable sound algorithms. We conclude with a list of open problems.

1.1 Related work

The containment problem in various XPath fragments has been a topic of wide interest for the past several years. A polynomial time algorithm for tree patterns without wildcard based on homomorphism between queries was given in [2]. The main result of Miklau and Suciu [12] is the coNP-completeness of containment where child, descendent, wildcard axes and filter expression are present. Almost a complete picture of the containment problem in the XPath fragments with disjunction, in the presence of DTDs and variables was given in [13]. Notably, it was shown that with a finite alphabet the containment problem rises to PSpace. [14] 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 [7, 8].

A closely related problem is XPath satisfiability [9, 3]. Query containment reduces to XPath satisfiability in fragments with enough expressive power (e.g. with negation and filter expressions). In [5], the query evaluation and satisfiability problems for Boolean combinations of tree patterns with equality and inequality constraints on data values were studied. Recently the containment of Boolean combinations of tree patterns was studied in [6].

A closely related work is [1], where the containment problem for tree patterns with general arithmetic comparisons was considered. In particular, their fragment is able to express data comparison of two different nodes. Afrati et al. show that containment in this fragment is Π_2^P-complete.

2. PRELIMINARIES

We work with node-labelled unranked finite trees, where the node labels are elements of an infinite set of tag names Σ. Formally, a tree over Σ is a tuple (N, E, r, ρ), where N, the set of nodes, is a prefix closed set of finite sequences of natural numbers, $E = \{(\langle n_1, \ldots, n_k\rangle, \langle n_1, \ldots, n_k, n_{k+1}\rangle) \mid \langle n_1, \ldots, n_{k+1}\rangle \in N\}$ is the child relation, $r = \langle\rangle$ is the root of the tree and ρ is the function assigning to each node in N a finite subset of Σ. Let A be an alphabet of attribute names. A tree with attributes is a tree extended with a partial function $att : N \times A \to D$, where D is a set of data values. When we make no restrictions, we assume that D is a dense linear order without endpoints. Trees in which $\rho(\cdot)$ is always a singleton are called *single-labelled* or *XML trees*. Trees without this restriction are called *multi-labeled trees*. By $T.n$ we denote the subtree of T rooted in n. We denote by E^+ the descendant relation, which is the transitive closure of the child relation E. A *path* from a node n to a node m is a sequence of nodes $n = n_0, \ldots, n_k = m$, with $k > 0$, such that for each $i \leq k$, $(n_i, n_{i+1}) \in E$.

Definition 1. (Tree Patterns with attribute value comparisons and label negation) Let $\neg\Sigma = \{\neg p \mid p \in \Sigma\}$. A tree pattern is a tuple $t = (N, E_/, E_{//}, r, o, \rho)$ such that $(N, E_/ \cup E_{//}, r, \rho)$ is a tree, where N is the set of nodes, $E_/, E_{//} \subseteq N^2$, such that $E_/ \cap E_{//} = \emptyset$, are the sets of child and descendent edges respectively, r is the root of the tree, o is the output node and ρ is the labeling function assigning to each node in N a finite set of labels from Σ, a finite set of labels from $\neg\Sigma$ and a finite set of value comparisons $@_a \texttt{ op } c$, where $a \in A, c \in D$ and $\texttt{op} \in \{=, \neq, <, >, \leq, \geq\}$. A tree pattern is *Boolean* if $o = r$.

The semantics of tree patterns is given in terms of embeddings.

Definition 2. (Embedding) Let $t = (N, E_/, E_{//}, r, o, \rho)$ be a tree pattern and $T = (N', E', r', \rho', att')$ a tree over Σ with attributes in A and values in D. A function $e : N \to N'$ is called an embedding of t into T if the following conditions are satisfied.

(i) Root preserving. $e(r) = r'$,
(ii) Edge preserving. For every $(n_1, n_2) \in E_/(E_{//})$, it holds that $(e(n_1), e(n_2)) \in E'(E'^+)$,
(iii) Label preserving. For every $n \in N$, if $p \in \rho(n)$ then $p \in \rho'(e(n))$ and if $\neg p \in \rho(n)$ then $p \notin \rho'(e(n))$,
(iv) Attribute comparison preserving. For every $n \in N$, if $@_a \texttt{ op } c \in \rho(n)$ then $att'(e(n), a) = c'$ and $D \models c' \texttt{ op } c$ for $\texttt{op} \in \{=, \neq, <, >, \leq, \geq\}$.

By $t(T)$ we denote the result of applying t to T, defined as $t(T) = \{e(o) \mid e \text{ is an embedding of } t \text{ into } T\}$.

Containment problem

Definition 3. Let t_1 and t_2 be two tree patterns. We say that t_1 is *contained* in t_2, notation $t_1 \subseteq t_2$, if for every tree T, $t_1(T) \subseteq t_2(T)$. Containment over single-labelled trees with attributes is denoted by $\subseteq$, and containment over multi-labelled attribute-free trees by $\subseteq_{\mathsf{ML}}$.

As usual (see [12, 10]), the tree pattern containment problem can be reduced to a containment problem of Boolean tree patterns only. Thus we will concentrate on studying the complexity of the containment problem of Boolean tree patterns only. We now give an equivalent definition of Boolean tree patterns via modal logic style formulas. Formulas of $\mathsf{TP}^@$ are defined by the following grammar.

$$\varphi \quad ::= \quad p \mid \top \mid @_a \texttt{ op } c \mid \varphi \wedge \varphi \mid \langle\downarrow\rangle\varphi \mid \langle\downarrow^+\rangle\varphi,$$

where $p \in \Sigma, a \in A, c \in D$ and $\texttt{op} \in \{=, \neq, <, >, \leq, \geq\}$. We then give the semantics for $\mathsf{TP}^@$ formulas. Let $T = (N, E, r, \rho, att)$ be a tree over Σ with attributes in A and values in D, and n a node in T.

- $T, n \models \top$,
- $T, n \models p$ iff $p \in \rho(n)$,
- $T, n \models @_a \texttt{ op } c$ iff $att(n, a) = c'$ and $D \models c' \texttt{ op } c$,
- $T, n \models \varphi \wedge \psi$ iff $T, n \models \varphi$ and $T, n \models \psi$,
- $T, n \models \langle\downarrow\rangle\varphi$ iff there is a node m with $(n, m) \in E$ and $T, m \models \varphi$,
- $T, n \models \langle\downarrow^+\rangle\varphi$ iff there is a node m with $(n, m) \in E^+$ and $T, m \models \varphi$.

Sometimes we write $T \models \varphi$ to denote $T, r \models \varphi$. We say that φ is *contained* in ψ if for every tree T and $n \in T$ we have $T, n \models \varphi$ implies $T, n \models \psi$. Given a containment problem $\varphi \subseteq \psi$ in a fragment of $\mathsf{TP}^@$, by Σ_p, Σ_a and Σ_c we denote respectively the sets of labels, attribute names and elements of D appearing in φ or ψ. Each tree pattern with attribute value comparisons can be transformed into a formula in $\mathsf{TP}^@$

and vice versa [12]. Let φ be in $\mathsf{TP}^{@}$, by $t(\varphi)$ we denote its corresponding tree pattern representation. Vice versa, if t is a tree pattern with attribute comparisons, by $\varphi(t)$ we denote the corresponding formula in $\mathsf{TP}^{@}$.

PROPOSITION 1. *Let $t = (N, E_{/}, E_{//}, r, r, \rho)$ be a Boolean tree pattern with comparisons, $\varphi(t)$ its formula representation in $\mathsf{TP}^{@}$, $T = (N', E', r', \rho', att')$ a tree over Σ, A and D, and n a node in T. Then there exists an embedding e from φ into $T.n$ if and only if $T, n \models \varphi$.*

Thus Boolean tree pattern containment can be reduced to containment of $\mathsf{TP}^{@}$ formulas.

By TP we denote tree patterns without attribute value comparisons. Furthermore, by TP^{C}_{O}, where $O \subset \{=, \neq, <, >, \leq, \geq\}$ and $C \subset \{@, \downarrow, \downarrow^{+}, \top\}$, the corresponding fragments of $\mathsf{TP}^{@}$ with the constructs from C and attribute value comparisons with operations from O. Here $\downarrow$, $\downarrow^{+}$ and $\top$ indicate the tree patterns allow child, descendent and wildcard constructs.

Expansions. By $\mathsf{TP}^{@,S}$, where $S \subseteq \{\vee, \neg^{g}\}$, we denote the formulas of $\mathsf{TP}^{@}$ extended by disjunction and guarded label negation, $p \wedge \neg q_1 \wedge \ldots \wedge \neg q_k$, where $p, q_1, \ldots, q_k$ are labels from Σ. By TP^{S}, $S \subseteq \{\vee, \neg^{g}\}$ we denote the attribute value comparisons free fragment of $\mathsf{TP}^{@,\vee,\neg^{g}}$. We assume $T, n \models p \wedge \neg q_1 \wedge \ldots \wedge \neg q_k$ iff $p \in \rho(n)$ and $q_i \notin \rho(n)$, $1 \leq i \leq k$.

Similarly to Lemma 3 in [12], we can prove the following proposition which is useful for our upper and lower bound proofs.

PROPOSITION 2. *Let $S = \emptyset$ or $S = \{\neg^{g}\}$. Let φ be a $\mathsf{TP}^{@,S}$ formula and Δ a finite set of $\mathsf{TP}^{@,S}$ formulas. Then there are* PTIME *computable $\mathsf{TP}^{@,S}$ formulas φ' and ψ' such that*

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

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

We define the translation $\widetilde{(\cdot)}$ which assigns a label to an attribute value comparison, $\widetilde{@_a \ \mathtt{op} \ c} = p_{@_a \ \mathtt{op} \ c}$. This mapping then can be homomorphically extended to the translation $\widetilde{(\cdot)}$ from formulas in $\mathsf{TP}^{@}$ over Σ, A and D to formulas without attribute value comparisons in TP over $\Sigma' = \Sigma \cup \{p_{@_a \ \mathtt{op} \ c} \mid \mathtt{op} \in \{=, \neq, <, >, \leq, \geq\}, a \in A, c \in D\}$.

3. CONTAINMENT FOR TREE PATTERNS WITH ATTRIBUTE VALUE COMPARISONS

In this section we first show that the containment problem in $\mathsf{TP}^{\vee,\neg^{g}}$ over multi-labeled trees is in coNP. Then using this fact we can show that the containment in $\mathsf{TP}^{@}$ is in coNP as well.

THEOREM 1. *The containment problem in $\mathsf{TP}^{\vee,\neg^{g}}$ over multi-labeled trees is in* coNP.

PROOF. The proof is similar to the arguments from [12] and [13]. Let φ be a $\mathsf{TP}^{\neg^{g}}$ formula. For simplicity, we use the same letter for a $\mathsf{TP}^{\neg^{g}}$ formula and its tree pattern representation. By $\sigma(\varphi)$ we denote the set of labels in Σ occurring in φ. Let $\Sigma_0 \subset \Sigma$ be a finite set of labels such that $\sigma(\varphi) \subseteq \Sigma_0$. Intuitively we define a canonical tree for φ as a tree obtained from the tree representation of φ by first replacing every descendent edge by a child-path where each node is labeled with a special symbol $\sharp$.

Let $\rho_N : N \to 2^{\Sigma_0 \cup \neg\Sigma_0}$ be the node labeling function of φ. We say that a function $\rho : N \to 2^{\Sigma_0}$ *positively extends* ρ_N if $\rho_N(v) \restriction_{\Sigma} \subseteq \rho(v)$ and $\rho(v)$ is consistent with $\rho_N(v)$ for every $v \in N$.

Now we define canonical models. Let $\{d_1, \ldots d_n\}$ be the descendant edges of φ. Given n non-negative numbers $\bar{u} = (u_1, \ldots, u_n)$ and $\rho : N \to 2^{\Sigma_0}$ which positively extends ρ_N, we define the $(\bar{u}, \rho)$-extension of φ, denoted as $\varphi[\bar{u}, \rho]$, as the tree pattern obtained by replacing each descendant edge d_i with a child-path of length u_i where each node is labeled by $\{\top\}$. Furthermore, the labeling in $\varphi[\bar{u}, \rho]$ is according to ρ.

Note that for any tree T and $\varphi \in \mathsf{TP}^{\neg^{g}}$, if there exists an embedding $e : \varphi \to T$, then there exist $\bar{u}$, and a unique embedding $e' : \varphi[\bar{u}, \rho] \to T$, where $\rho = \rho_T \restriction_{\Sigma_0}$ (ρ_T is the labeling function of T), such that e' extends e.

A *canonical tree* $t(\varphi[\bar{u}, \rho])$ is the tree obtained from $\varphi[\bar{u}, \rho]$ by changing the labels of the nodes labeled by $\{\top\}$ to $\sharp$. Such nodes labeled with $\sharp$ are called *special*.

We define the $\top$-length of a tree pattern $\psi \in \mathsf{TP}^{\neg^{g}}$ as the largest number k such that there exist k nodes $v_1, \ldots, v_k$ connected by child edges and $\rho(v_i) = \{\top\}$ in ψ. The coNP upper bound directly follows from the next lemma.

LEMMA 1. *Let φ and ψ be in $\mathsf{TP}^{\vee,\neg^{g}}$, $\Sigma_0 = \sigma(\varphi) \cup \sigma(\psi)$. Then $\varphi \not\subseteq_{\mathsf{ML}} \psi$ if and only if there exists a tree T over Σ_0 such that $T \models \varphi$ and $T \not\models \psi$ and the size of T is polynomial in the size of φ and ψ.*

PROOF OF LEMMA. The direction ($\Leftarrow$) is obvious.

($\Rightarrow$) Assume there exists a tree T with $T \models \varphi$ and $T \not\models \psi$. W.l.o.g. we can assume that the label sets of T are subsets of $\Sigma_0 \subset \Sigma$, the set of labels occurring in φ or ψ. Let $\bigvee_i \varphi_i$ and $\bigvee_j \psi_j$ be the DNFs of φ and ψ.

Since $T \models \varphi$, there exists an embedding $e : \varphi_i \to T$ for some i. Let e' be the corresponding embedding of $\varphi_i[\bar{u}, \rho]$ into T, where ρ is the labeling function of T. Let T_1 be the canonical tree $t(\varphi_i[\bar{u}, \rho])$. Note that the number of nodes in T_1 not labeled with $\sharp$ is at most the number of nodes in φ_i.

We show that $T_1 \not\models \psi$. Suppose the opposite, i.e. there exists an embedding $e_1 : \psi_j \to T_1$ for some j. We then define the function $f : \psi_j \to T$ by composing e_1 and e'. The function f preserves the structure, since T_1 and $\varphi_i[\bar{u}, \rho]$ have the same structure and e_1 and e' are embeddings. Moreover, f preservers the labels. Let v be a node in ψ_j. We consider two cases:

- $p \in \rho_N(v)$. Then p is in the label of $e_1(v)$ in T_1. In particular, $e_1(v)$ is not a special node, i.e. not labeled with $\sharp$. Thus, $e_1(v) \in dom(e')$ and, therefore, p is in the label of $f(v)$.
- $\neg p \in \rho_N(v)$. As $\psi_j \in \mathsf{TP}^{\neg^{g}}$, there exists a label $q \in \rho(v)$ and, thus, $e_1(v)$ is not a special node. Thus we have that p is not in the label of $e_1(v)$, as e_1 is an embedding. Since the labeling of the non-special nodes in T_1 and T coincide, we have that p is not in the label of $f(v)$ either.

Hence, we have $T \models \psi_j$, as f is an embedding from ψ_j into T, which is a contradiction. Thus, $T_1 \not\models \psi$.

Note that T_1 is not yet our desired tree of polynomial size, since the paths of special nodes in T_1 might be too long. However, we can shorten them. We define the tuple of non-negative numbers $\bar{v} = (v_1, \ldots, v_n)$ as $v_i = min(u_i, k+1)$, where k is the $\top$-length of ψ_j. Then the canonical tree $T_2 := t(\varphi_i[\bar{v}, \rho])$ is of polynomial size in the size φ and ψ. We can show that still $T_2 \not\models \psi_j$. For this we need the following claim.

CLAIM 1. *Let a singleton path be a path in which each node, but the last one, has exactly one child. Let T be a tree such that $T \not\models \psi$ for $\psi \in \mathsf{TP}^{\neg^{g}}$, and let k be the maximal*

$\top$-length of ψ. Let also $v_1, \ldots, v_l$ be a singleton path in T labeled with $\sharp$ and such that $l > k+1$. Let T' be the tree obtained from T by deleting the node v_l and adding its children to v_{l-1}. Then still $T' \not\models \psi$.

The intuition for the proof of this claim lies in the fact that if there was an embedding of ψ into T', then it could be extended to an embedding into T, as there must be a descendant edge which can be mapped on such a long singleton path of special nodes in T'.

Now this claim can be used to show that $T_2 \not\models \psi_j$. Recall that $T_1 = t(\varphi_i[\bar{u}, \rho]) \not\models \psi_j$. Applying the claim $u_i - (k+1)$ times for the ith singleton path of special nodes, we get that $T_2 \not\models \psi_j$. Furthermore, T_2 is of polynomial size in the size of φ and ψ. □

3.1 Attribute value comparisons over dense unbounded order

We now show that the containment problem in $\mathsf{TP}^{@}$ over trees with attributes can be reduced in PTime to the containment problem in $\mathsf{TP}^{\vee,\neg^g}$ over multi-labeled attribute-free trees. Thus the containment for $\mathsf{TP}^{@}$ is in coNP as well. Here we make an assumption that the domain of attribute values is a dense linear order. The main result of this section is the following.

Theorem 2. *Let $S = \{\vee, \neg^g\}$. The containment problem in $\mathsf{TP}^{@,S}$ over trees with attributes is in* coNP.

Given the containment problem $\varphi \subseteq \psi$ for $\varphi, \psi \in \mathsf{TP}^{@,S}$, we reduce it to the containment problem $\varphi' \subseteq_{\mathsf{ML}} \psi'$ in $\mathsf{TP}^{\neg^g,S}$, which is in coNP by Theorem 1. Thus Theorem 2 is a consequence of the following lemma.

Lemma 2. *Let $S \subseteq \{\vee, \neg^g\}$ and φ, ψ be $\mathsf{TP}^{@,S}$ formulas. Then there exist* PTime *computable $\mathsf{TP}^{\neg^g,S}$ formulas φ' and ψ' such that*

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

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

Proof. We take $\varphi' := \widetilde{\varphi}$ and $\psi' := \widetilde{\psi} \vee Ax$, where $\widetilde{(\cdot)}$ was defined in Section 2 and Ax is the disjunction of the formulas in Figure 2. There we use the abbreviation $\langle\downarrow^*\rangle\theta = \theta \vee \langle\downarrow^+\rangle\theta$. Note that the formula Ax is in $\mathsf{TP}^{\vee,\neg^g}$. We then can show the following.

Claim 2. *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 $p_{@_a\ \mathtt{op}\ c} \in \rho(n)$ for every $\mathtt{op} \in \{=, \neq, \geq, \leq, <, >\}$,

(ii) there is exactly one $p_{@_a = c} \in \rho(n)$ and for every $c_1 \in \Sigma_c$ it holds that $p_{@_a\ \mathtt{op}\ c_1} \in \rho(n)$ iff $D \models c\ \mathtt{op}\ c_1$,

(iii) there is no $p_{@_a = c} \in \rho(n)$ and there exists $c' \in D \setminus \Sigma_c$ such that for every $c_1 \in \Sigma_c$ it holds that $p_{@_a\ \mathtt{op}\ c_1} \in \rho(n)$ iff $D \models c'\ \mathtt{op}\ c_1$.

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 $T, r \not\models \psi'$. W.l.o.g we can assume that T is defined over Σ'. Then we define a single-labeled tree $T' := (N, E, r, l, att)$, where l is the labeling function and att is a partial function assigning a value in D to a given node and an attribute name, as follows:

- For $p \in \Sigma_p$, $l(n) = p$ iff $p \in \rho(n)$. If there is no $p \in \Sigma_p$ such that $p \in \rho(n)$, we set $l(n) = z$ for a fresh symbol z.
- $$att(n,a) = \begin{cases} \text{undefined} & \text{if there is no } p_{@_a\ \mathtt{op}\ c_1} \text{ in } \rho(n), \\ c & \text{if } p_{@_a = c} \in \rho(n), \\ c' & \text{from Claim 2, (iii), otherwise.} \end{cases}$$

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

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

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

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

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

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

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

For every $a \in \Sigma_a, c, c_1, c_2 \in \Sigma_c$ and $R, S \in \{\neq, \leq, \geq, <, >\}$ with $R \neq S$,

$$\langle\downarrow^*\rangle(p_{@_a R c_1} \wedge \neg p_{@_a = c} \wedge \neg p_{@_a > c_2} \wedge \neg p_{@_a < c_2}), \qquad (\textit{DNeg})$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Figure 2: The disjuncts of the formula Ax from Lemma 2

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 2. By induction, using Claim 2, we can show that for every $\theta \in \mathsf{TP}^{@,S}$, $T, n \models \widetilde{\theta}$ iff $T', n \models \theta$. Thus, it follows $T', r \models \varphi$ and $T', r \not\models \psi$ which was desired.

($\Leftarrow$) Let $T = (N, E, r, l, att)$ 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 $p = l(n)$,
- $p_{@_a = c} \in \rho(n)$ iff $att(n,a) = c$,
- $p_{@_a\ \mathtt{op}\ c} \in \rho(n)$ iff $att(n,a) = c_1$ and $D \models c_1\ \mathtt{op}\ c$ for $\mathtt{op} \in \{\neq, \leq, \geq, <, >\}$, $c \in \Sigma_c$.

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

Now, if $\vee \in S$, we are done. In the remaining cases we apply Proposition 2 to remove the disjunctions in Ax where needed. Finally, in case of multi-labeled trees we do not include formulas (*Label*) as the disjuncts of Ax. □

3.1.1 Lower bound

We now give a lower bound for a small fragment of $\mathsf{TP}^{@}$.

PROPOSITION 3. *The containment problem in* $\mathsf{TP}^{@,\downarrow,\downarrow^+}_{=,\neq}$ *is* coNP*-hard.*

PROOF. We reduce a 3SAT problem to a non-containment problem in $\mathsf{TP}^{@,\downarrow,\downarrow^+}_{=,\neq}$.

Firstly, we can use disjunction of tree patterns on the right side of the containment problem, due to Proposition 2.

Let Q be the conjunction of clauses $C_i = (X^i_1 \vee X^i_2 \vee X^i_3), 1 \leq i \leq k$ over the variables $\{x_1, \ldots, x_n\}$, where X^i_j 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 $\varphi := r \wedge \langle\downarrow\rangle(b \wedge @_{a_1} \neq 2 \wedge \ldots \wedge @_{a_n} \neq 2)$ and $\psi := \bigvee_{i=1}^{k}\langle\downarrow\rangle(b \wedge B^i_1 \wedge B^i_2 \wedge B^i_3)$, where $B^i_j = (@_{a_l} = 0)$ iff $X^i_j = x_l$ in C_i and $B^i_j = (@_{a_l} \neq 0)$ iff $X^i_j = \neg x_l$ in C_i.

We claim that Q is satisfiable if and only if $\varphi \not\subseteq \psi$. □

3.2 Restricting the domain of attribute values

Theorem 2 was proved under the assumption that the domain for attribute values is dense unbounded linear order. In fact, if we further restrict the domain, the coNP upper bound for containment still holds in these cases.

PROPOSITION 4. *Let* $S = \{\vee, \neg^g\}$ *and* D *be a linear order such that it is one of the following:*

(i) finite,

(ii) discrete,

(iii) dense or discrete with one or two endpoints.

Then the containment problem in $\mathsf{TP}^{@,S}$ *over single-labeled trees with the domain of attribute values* D *is in* coNP.

PROOF. (Sketch) All the items can be proved using a variant of Lemma 2. That is, we reduce in PTIME a given containment problem $\varphi \subseteq \psi$ to the containment in $\mathsf{TP}^{\neg^g,S}$ over multi-labeled attribute-free trees. The reduction has the form $\varphi' := \widetilde{\varphi}$ and $\psi' := \widetilde{\psi} \vee Ax \vee Ax_k$, where Ax is in Figure 2 and $Ax_k, k \in \{(Fin), (Discr), (End)\}$ is constructed according to the cases.

In case the domain of attribute values D is finite, we take $Ax_{(Fin)}$ as the disjunction of the formulas: for every $a \in \Sigma_a$, $c \in \Sigma_c$ and $\mathtt{op} \in \{=, \neq, <, >, \leq, \geq\}$:

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

If D is discrete, then $Ax_{(Discr)}$ is the disjunction of the formulas: for every $a \in \Sigma_a, c_1, c_2 \in \Sigma_c$ such that $c_1 < c_2$ in D and there is no c' in D with $c_1 < c' < c_2$,

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

Finally, let D be dense or discrete with one or two endpoints. If D is dense, take $Ax_{(End)}$ as the disjunction of Ax from Figure 2 and the following formulas:

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

$$\langle\downarrow^*\rangle p_{@_a<c_l}. \qquad (LEnd)$$

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

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

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

3.3 Required attributes

In Section 3 we dealt with the case when attributes are optional. We now consider the cases when some attributes are required. We say that an attribute $a \in A$ is required if for every tree T and node $n \in T$, the function $att : N \times \{a\} \to D$ is total.

THEOREM 3. *The containment problem in* $\mathsf{TP}^{@}$ *over trees with at least one required attribute is* PSPACE*-complete.*

PROOF. For the upper bound, we reduce the containment problem in $\mathsf{TP}^{@}$ with required attributes to the implication problem in ∃CTL which is known to be in PSPACE, [11]. As the first step, we reduce the containment problem in $\mathsf{TP}^{@}$ to the containment in $\mathsf{TP}^{\neg}$ (tree pattern formulas with label negation) similar to Lemma 2. The additional axiom in Ax (Figure 2) is $\langle\downarrow^*\rangle(\neg p_{@_a=c} \wedge \neg p_{@_a\neq c})$ for every required $a \in A$ and $c \in \Sigma_c$. This is to enforce that a is defined everywhere in the tree. As the second step we translate the containment in $\mathsf{TP}^{\neg}$ to the implication problem in ∃CTL. We omit further details due to the lack of space.

For proving the lower bound we encode the corridor tiling problem, which is known to be hard for PSPACE [4]. Our lower bound proof uses the construction from the PSPACE-hardness proof for the containment problem in TP with disjunction over a finite alphabet in [13].

The corridor tiling problem is formalized as follows. Let $\mathsf{Til} = (D, H, V, \bar{b}, \bar{t}, n)$ 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, $\bar{b}$ and $\bar{t}$ 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 $\bar{b}$ and $\bar{t}$ respectively.

Let $a \in A$ be a required attribute. Now we construct two $\mathsf{TP}^{@}_{=,\neq}$ 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 the 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. In particular $u_1 = \bar{b}$ and $u_k = \bar{t}$.

For the sake of readability, we use the following abbreviation. Expressions φ_1/φ_2 and $\varphi_1//\varphi_2$ denote $\varphi_1 \wedge \langle\downarrow\rangle\varphi_2$ and $\varphi_1 \wedge \langle\downarrow^+\rangle\varphi_2$ respectively. Furthermore, $\varphi_1/^1\varphi_2/^2\ldots/^{l-1}\varphi_l$, where $/^i \in \{/, //\}$ must be read as $\varphi_1/^1(\varphi_2/^2(\ldots/^{l-1}\varphi_l)\ldots))$. We also say that, for expression r, r^i denotes $r/\ldots/r$ with i occurrences of r. We then define the formulas over attributes $\{a\}$ and attribute domain containing $D \cup \{\sharp\}$.

- $\bar{t} = @_a = t_1/@_a = t_2/\ldots/@_a = t_n/@_a = \#$,
- $\bar{b} = @_a = b_1/\ldots/@_a = b_n/@_a = \#$.

Let then define $\varphi' := \bar{b}//\bar{t}$. Intuitively, this expression enforces a tiling to start with a path starting with $\bar{b}$ and finishing with $\bar{t}$. Now the formula ψ' defines all incorrect tilings and additional constraints. It is the disjunction of the following $\mathsf{TP}^{@,\vee}_{=,\neq}$ formulas.

- $\bigvee_{i=0}^{n-1}\bar{b}//@_a = \sharp/\top^i/@_a = \sharp$ a row is too short,
- $\bar{b}//(@_a \neq \sharp)^{n+1}$ a row is too long,
- $\bigvee_{d\in D}\bar{b}//(@_a = d \wedge @_a = \sharp)$, a tile and the delimiter occur at the same time,
- $\bigvee_{d_i,d_j\in D,\ i\neq j}\bar{b}//(@_a = d_i \wedge @_a = d_j)$, there are two tiles on a position,
- $\bar{b}//(@_a \neq d_1 \wedge \ldots \wedge @_a \neq d_m \wedge @_a \neq \sharp)$, neither the delimiter or a tile on a position,

- $\bigvee_{(d_1,d_2)\notin H} \bar{b}//@_a = d_1/@_a = d_2$, a horizontal constraint is violated,
- $\bigvee_{(d_1,d_2)\notin V} b_1//@_a = d_1/\top^n/@_a = d_2$, a vertical constraint is violated.

We then apply Proposition 2 to remove the outermost disjunction in ψ' to obtain the equivalent containment problem $\varphi \subseteq \psi$ in $\mathsf{TP}^{@}_{=,\neq}$. □

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

PROPOSITION 5. *The containment problem in* $\mathsf{TP}^{@}$ *with required attributes at elements is in* coNP.

PROOF. As before, we can prove a variant of Lemma 2. In this case we take $\varphi' := \widetilde{\varphi}$ and ψ' as the disjunction of $\widetilde{\psi}$, Ax (from Figure 2) and $\langle\downarrow^*\rangle(p \wedge \neg p_{@_a=c} \wedge \neg p_{@_a\neq c})$ for every $c \in \Sigma_c$ and $a \in \Sigma_a$ required at element $p \in \Sigma_p$. The axiom enforces the requirement that every node with p-label must have a value for a-attribute. □

4. TRACTABLE FRAGMENTS

In this section we consider fragments of tree patterns with attributes value comparisons where the containment problem remains in PTime. It is known that containment in $\mathsf{TP}^{\downarrow,\top}$ and $\mathsf{TP}^{\downarrow,\downarrow^+}$ is decidable in PTime, [2, 12].

PROPOSITION 6. *Let* TP^{X} *be any fragment whose containment problem over multiple-labeled trees is in* PTime. *Then the containment problem in* $\mathsf{TP}^{@,X}_{=}$ *over multi-labeled trees with attributes is also in* PTime.

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

Our algorithm first checks (in PTime) if φ is consistent, i.e. if it contains both $@_a = c$ and $@_a \neq c$ or both $@_a = c$ and $@_a = d$ in the label of a node in $t(\varphi)$ 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 2 by reduction to a containment of attribute-free formulas using the translation $\tilde{(\cdot)}$ and the formula $(Label)$ only. □

The rewriting technique from the last proof can also be applied on TP fragments with $=$ and $\neq$ but then it yields only a sound algorithm. For $\mathsf{TP}^{\downarrow,\downarrow^+}_{=,\neq}$, this algorithm, assuming PTime $\neq$ NP, must be incomplete by Proposition 3. For $\mathsf{TP}^{\downarrow,\top}_{=,\neq}$ and $\mathsf{TP}^{\downarrow^+,\top}_{=,\neq}$ it is open whether this algorithm is complete.

PROPOSITION 7. *Let* $\mathsf{TP}^{@,X}_{=,\neq}$ *be a tree pattern fragment, and* TP^{X} *the corresponding fragment without attribute-value comparisons. For consistent* φ *and* ψ*, it holds that* $\tilde{\varphi} \subseteq \tilde{\psi}$ *implies* $\varphi \subseteq \psi$.

PROOF. Let $T = (N, E, r, l, att)$ be a tree such that $T \models \varphi$ and $T \not\models \psi$. We then define a tree $T' = (N, E, r, \rho)$, where the labeling function ρ is defined as follows.

$$\begin{array}{rcl} \rho(n) & = & \{l(n)\} \cup \{p_{@_a=c} \mid att(n,a) = c\} \cup \\ & \cup & \{p_{@_a\neq c_1} \mid c_1 \in \Sigma_c, att(n,a) \neq c_1\}. \end{array}$$

We claim that $T' \models \tilde{\varphi}$ and $T' \models \tilde{\psi}$. □

5. CONCLUSION

We showed that optional attribute value comparisons using all XPath operators do not increase the complexity of the containment problem when added to tree patterns with child, descendant and wildcard. For the PTime TP fragment with child and descendant, we showed that adding equality and inequality comparisons causes an increase of complexity to coNP. For the other PTime fragments studied in [2, 12], (i.e., wildcard and one of child and descendant), the upper bound is still open. For these fragments, we presented a PTime algorithm which is complete for input with only equality comparisons, but only known to be sound for equality and inequality comparisons.

The containment problem for TP with global comparisons studied in [1] was shown to be Π_2^P-hard already with only equality and inequality, and in coNP with only equality. The exact complexity with just inequality comparisons remains open.

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

6. REFERENCES

[1] F. N. Afrati, S. Cohen, and G. M. Kuper. On the complexity of tree pattern containment with arithmetic comparisons. *Inf. Process. Lett.*, 111(15):754–760, 2011.

[2] S. Amer-Yahia, S. Cho, L. Lakshmanan, and D. Srivastava. Tree pattern query minimization. *The VLDB Journal*, 11:315–331, 2002.

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

[4] B. S. Chlebus. Domino-tiling games. *J. Comput. Syst. Sci.*, 32(3):374–392, 1986.

[5] C. David. Complexity of data tree patterns over XML documents. In E. Ochmanski and J. Tyszkiewicz, editors, *MFCS*, volume 5162 of *Lecture Notes in Computer Science*, pages 278–289. Springer, 2008.

[6] C. David, A. Gheerbrant, L. Libkin, and W. Martens. Containment of pattern-based queries over data trees. In *ICDT*, 2013.

[7] A. Deutsch and V. Tannen. Containment and integrity constraints for XPath. In M. Lenzerini, D. Nardi, W. Nutt, and D. Suciu, editors, *KRDB*, volume 45 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2001.

[8] A. Deutsch and V. Tannen. XML queries and constraints, containment and reformulation. *Theor. Comput. Sci.*, 336(1):57–87, 2005.

[9] J. Hidders. Satisfiability of XPath expressions. In G. Lausen and D. Suciu, editors, *DBPL*, volume 2921 of *Lecture Notes in Computer Science*, pages 21–36. Springer, 2003.

[10] B. Kimelfeld and Y. Sagiv. Revisiting redundancy and minimization in an XPath fragment. In *EDBT'08*, pages 61–72.

[11] O. Kupferman and M. Y. Vardi. An automata-theoretic approach to modular model checking. *ACM Trans. Prog. Lang. Syst.*, 22(1):87–128, 2000.

[12] G. Miklau and D. Suciu. Containment and equivalence for a fragment of XPath. *J. ACM*, 51(1):2–45, 2004.

[13] F. Neven and T. Schwentick. On the complexity of XPath containment in the presence of disjunction, DTDs, and variables. *Logical Methods in Computer Science*, 2(3), 2006.

[14] P. T. Wood. Containment for XPath fragments under DTD constraints. In *ICDT 2003*, pages 297–311.