The Consistency and Absolute Consistency Problems of XML Schema Mappings between Restricted DTDs

Hayato Kuwada[1], Kenji Hashimoto[2], Yasunori Ishihara[1], and Toru Fujiwara[1]

[1] Osaka University, Japan
{h-kuwada,ishihara,fujiwara}@ist.osaka-u.ac.jp
[2] Nara Institute of Science and Technology, Japan
k-hasimt@is.naist.jp

Abstract. Consistency of XML schema mappings, which means that some document conforming to the source schema can be mapped into a document conforming to the target schema, is an essentially necessary property. It is also important for XML schema mappings to be absolutely consistent, that is, every document conforming to the source schema can be mapped into a document conforming to the target schema. As a known result, consistency of a mapping between general DTDs is EXPTIME-complete even if the class of document patterns for defining mappings is restricted to downward axes and qualifiers. In addition, the known tractability result is only on a restricted class of document patterns under restricted DTDs called nested-relational DTDs. Moreover, there are few known results on the tractability of absolute consistency. In this paper, we discuss the consistency and absolute consistency problems under restricted DTDs called disjunction-capsuled DTDs, which were proposed by Ishihara et al. We show that for many document pattern classes, both problems are solvable in polynomial time under disjunction-capsuled DTDs. Although disjunction-capsuled DTDs are an incomparable class to nested-relational DTDs, a part of our results can be extended to a proper superclass of nested-relational DTDs.

1 Introduction

XML (Extensible Markup Language) is a markup language for describing structured data and used for a variety of applications and database systems. Accordingly, many schemas are defined to specify document structures, and schema description languages such as DTD (Document Type Definition) are used for specifying schemas. A schema mapping is the basis of data transformation from one schema to another caused by data exchange or schema evolution. In particular, schema mappings for XML are called *XML schema mappings*. A mapping is a triple of a *source schema*, a *target schema*, and a set of *dependencies*, which are pairs of *document patterns* and specify correspondence of data. *Consistency* of schema mappings, which means that some document conforming to the source

Y. Ishikawa et al. (Eds.): APWeb 2013, LNCS 7808, pp. 228–239, 2013.

schema can be mapped into a document conforming to the target schema according to the given dependencies, is an essentially necessary property. It is also important for schema mappings to be *absolutely consistent*, that is, every document conforming to the source schema can be mapped into a document conforming to the target schema according to the given dependencies.

Example 1. Define the source and target schemas D_S and D_T as follows:

D_S:	D_T:
`root -> student`$^+$	`root -> student`$^+$
`student -> name phone`	`student -> name phone`*
`phone -> home mobile other`*	`phone -> home | mobile | other`

First, consider the following dependency set Σ_1:

$$\Sigma_1 = \{\texttt{student/phone[home][mobile]} \longrightarrow \texttt{student/phone[home][mobile]}\}.$$

Σ_1 states that if there is a document of D_S such that a `phone` node has both `home` and `mobile` nodes as its children, then the document is mapped into a document of D_T with a `phone` node which has, again, both `home` and `mobile` nodes as its children. However, Σ_1 is not consistent because in any document of D_T, a `phone` node cannot have both `home` and `mobile` nodes as its children. Next, consider the following dependency set Σ_2:

$$\Sigma_2 = \{\texttt{student/phone[home][other]} \longrightarrow \texttt{student/phone[home][other]}\}.$$

Σ_2 is consistent because there is a document of D_S which has no `other` nodes. Such a document vacuously satisfies Σ_2. On the other hand, Σ_2 is not absolutely consistent because of the similar reason to the case of Σ_1. Lastly, consider the following dependency set Σ_3:

$$\Sigma_3 = \{\texttt{student/phone[home][mobile][other]} \longrightarrow \texttt{student[phone/home][phone/mobile][phone/other]}\}.$$

Σ_3 is absolutely consistent because for every document of D_S, if it has a `phone` node with children `home`, `mobile`, and `other`, then it can be mapped into a document of D_T which has (at least three) `phone` nodes one of which has `home`, another has `mobile`, and the other has `other` as their children.

Amano et al. investigated the computational complexity of the problems of determining consistency and absolute consistency of XML schema mappings [1]. The main focus of [1] is XML schema mappings with data values, but they addressed also mappings without data values. The results of the case without data values are shown in Table 1. In this paper we use XPath expressions for specifying document patterns, although in [1] tree patterns is used. Using XPath expressions allows us to utilize the rich known results on the satisfiability problem for our problem. XPath expressions consist of $\downarrow$ (child axis), $\downarrow^*$ (descendant-or-self axis), $\uparrow$ (parent axis), $\uparrow^*$ (ancestor-or-self axis), $\rightarrow$ (next-sibling axis), $\rightarrow^*$ (following-sibling-or-self axis), $\rightarrow^+$ (following-sibling axis), $\leftarrow^+$ (preceding-sibling axis),

Table 1. Known results in [1]

XPath class											General DTDs	
$\downarrow$	$\downarrow^*$	$\uparrow$	$\uparrow^*$	$\rightarrow$	$\rightarrow^*$	$\rightarrow^+$	$\leftarrow^+$	$\cup$	[ ]	$*$	Consistency	Absolute Consistency
+	+								+	+	EXPTIME-complete	Π_2^p-complete
+	+			+	+				+	+	EXPTIME-complete	Π_2^p-complete

XPath class											Nested-relational DTDs	
$\downarrow$	$\downarrow^*$	$\uparrow$	$\uparrow^*$	$\rightarrow$	$\rightarrow^*$	$\rightarrow^+$	$\leftarrow^+$	$\cup$	[ ]	$*$	Consistency	Absolute Consistency
+									+		PTIME	PTIME
+	+								+	+	PTIME	Π_2^p
+	+			+					+		PSPACE-hard	Π_2^p

Table 2. Results of this paper

XPath class											DC-DTDs	
$\downarrow$	$\downarrow^*$	$\uparrow$	$\uparrow^*$	$\rightarrow$	$\rightarrow^*$	$\rightarrow^+$	$\leftarrow^+$	$\cup$	[ ]	$*$	Consistency	Absolute Consistency
+	+					+	+	+	+	+	PTIME	PTIME
+	+	+	+			+	+	+		+	PTIME	PTIME
+		+				+			+		PTIME	PTIME

$\cup$ (path union), [ ] (qualifier), and $*$ (wild card). Table 1 describes computational complexity of each class including only axes and operators with the "+". Nested relational DTDs are non-recursive DTDs such that each content model is in the form of $\hat{l}_1 \cdots \hat{l}_m$, where all l_i's are distinct and each $\hat{l}_i$ is either l_i, ${l_i}^*$, ${l_i}^+$, or ${l_i}^?$.

As can be seen from Table 1, constraints under which decision problems of consistency and absolute consistency are solved in PTIME have been little known, even if data values are not concerned. Thus, the purpose of this paper is to find practically wide classes of DTDs and XPath expressions for which both problems can be solved in PTIME. As a first step for achieving this, we focus on the case without data values in this paper.

In this paper, we use a restricted class of DTDs called disjunction-capsuled DTDs (DC-DTDs), which were proposed in [2]. DC-DTDs are an important class from the theoretical point of view because many tractability results on XPath satisfiability under DC-DTDs are known. Such results can be exploited for consistency and absolute consistency problems. DC-DTDs are also a meaningful class from the practical point of view, because 7 out of 22 real-world DTDs, 853 out of 950 real-world DTD rules are actually DC [3]. Although DC-DTDs are incomparable to nested-relational DTDs, the tractability results on the satisfiability under DC-DTDs can be extended to a wider class, called DC$^{?+}$-DTDs [3],

which is a proper superclass of nested-relational DTDs. 13 out of 22 DTDs, 928 out of 950 rules are $DC^{?+}$.

The results of this paper are shown in Table 2. "+" acrossing multiple cells represents "one of them". First, it is shown that consistency of mappings between DC-DTDs is solvable by deciding satisfiability and validity of document patterns *linearly many times*. Precisely speaking, we show that under DC-DTDs, satisfiability of a set of dependencies can be decomposed into satisfiability and validity of each document pattern in the dependencies. Hence, if the document patterns are in an XPath class for which satisfiability and validity are tractable, then consistency is also tractable. Next, it is shown that absolute consistency of mappings between DC-DTDs is solvable by deciding satisfiability of document patterns *linearly many times*. Again, the technically important point is that satisfiability of a set of dependencies can be done by independently checking the satisfiability of each document pattern in the dependencies. Hence, if the document patterns are in an XPath class for which satisfiability is tractable, then absolute consistency is also tractable.

The rest of this paper is organized as follows. Several definitions are given in Section 2. Sections 3 and 4 present tractability results of consistency and absolute consistency of mappings between DC-DTDs, respectively. Section 5 summarizes the paper.

2 Preliminaries

2.1 XML Documents and DTDs

In this paper, we regard elements of XML documents as labeled nodes, and entire XML documents as labeled ordered trees. In what follows, we write trees to mean labeled ordered trees. Let $\lambda(v)$ denote the label of a node v. For a sequence of nodes $v_1 v_2 \cdots v_n$, define $\lambda(v_1 v_2 \cdots v_n)$ as $\lambda(v_1)\lambda(v_2)\cdots\lambda(v_n)$.

A regular expression over Γ consists of ϵ (empty sequence), the symbols in Γ, and operators $\cdot$ (concatenation, usually omitted in the notation), $|$ (disjunction), and $*$ (repetition). We exclude $\emptyset$ (empty set) because we are interested in only nonempty regular expressions. Let $L(e)$ denote the string language represented by a regular expression e.

Definition 1. *A* DTD D *is a triple* (Γ, r, P)*, where*

- Γ *is a finite set of labels,*
- $r \in \Gamma$ *is the root label, and*
- P *is a mapping from* Γ *to the set of regular expressions over* Γ*.* $P(l)$ *is called the* content model *of label* l*.*

A tree T conforms *to a DTD* $D = (\Gamma, r, P)$ *if the root label of* T *is* r*, and for each node* v *of* T *and its children sequence* $v_1 v_2 \cdots v_n$*, it holds that* $\lambda(v_1 v_2 \cdots v_n) \in L(P(\lambda(v)))$*. Let* $TL(D)$ *denote the set of all the trees conforming to* D*.*

Next, we introduce DC-DTDs, a restricted class of DTDs.

Definition 2. *A regular expression e is* disjunction-capsuled *[2], or* DC *for short, if e is in the form of $e_1 e_2 \cdots e_n$ ($n \geq 1$), where each e_i is either a symbol in Γ or in the form of $(e'_i)^*$ for an arbitrary regular expression e'_i.*

For example, $(a|b^*)^*ba^*$ is disjunction-capsuled, and ϵ is also disjunction-capsuled because ϵ is ϵ^*. On the other hand, $a^*|b^*$ is not disjunction-capsuled.

Definition 3. *Let $D = (\Gamma, r, P)$. D is a* disjunction-capsuled DTD*, or* DC-DTD *for short, if $P(l)$ is disjunction-capsuled for each $l \in \Gamma$.*

2.2 XPath Expressions

XPath is a query language for XML documents.

Definition 4. *The syntax of an XPath expression p is defined as follows:*

$$
\begin{aligned}
p &::= \chi :: l \mid p/p \mid p \cup p \mid p[q], \\
\chi &::= \cdot \mid \downarrow \mid \uparrow \mid \downarrow^* \mid \uparrow^* \mid \rightarrow^+ \mid \leftarrow^+, \\
q &::= p \mid q \wedge q \mid q \vee q,
\end{aligned}
$$

where $l \in \Gamma$. Let $\mathcal{X}$ denote the set of all XPath expressions.

Note that $\rightarrow$ (next-sibling axes) and $\rightarrow^*$ (following-sibling-or-self axes) are excluded from $\mathcal{X}$ because they are not proper axes in W3C XPath. Moreover, $*$ (wild card) is also excluded because it can be represented by using $\cup$ (path union). On the other hand, $\mathcal{X}$ contains $\cdot$ (self axis), which is not considered in previous research. Self axis makes our discussion simpler without destroying the known tractability results of XPath satisfiability.

Definition 5. *The semantics of an XPath expression p over a tree T is defined as follows, where p and q are regarded as binary and unary predicates over nodes of T, respectively:*

- $T \models (\cdot :: l)(v, v)$ *if* $\lambda(v) = l$;
- $T \models (\downarrow :: l)(v, v')$ *if v' is a child of v and* $\lambda(v') = l$;
- $T \models (\uparrow :: l)(v, v')$ *if v' is a parent of v and* $\lambda(v') = l$;
- $T \models (\downarrow^* :: l)(v, v')$ *if v' is v or a descendant of v, and* $\lambda(v') = l$;
- $T \models (\uparrow^* :: l)(v, v')$ *if v' is v or an ancestor of v, and* $\lambda(v') = l$;
- $T \models (\rightarrow^+ :: l)(v, v')$ *if v' is a following sibling of v and* $\lambda(v') = l$;
- $T \models (\leftarrow^+ :: l)(v, v')$ *if v' is a preceding sibling of v and* $\lambda(v') = l$;
- $T \models (p/p')(v, v')$ *if there is v'' such that* $T \models p(v, v'')$ *and* $T \models p'(v'', v')$;
- $T \models (p \cup p')(v, v')$ *if* $T \models p(v, v')$ *or* $T \models p'(v, v')$;
- $T \models (p[q])(v, v')$ *if* $T \models p(v, v')$ *and* $T \models q(v')$;
- $T \models p(v)$ *if there is v' such that* $T \models p(v, v')$;
- $T \models (q \wedge q')(v)$ *if* $T \models q(v)$ *and* $T \models q'(v)$;
- $T \models (q \vee q')(v)$ *if* $T \models q(v)$ *or* $T \models q'(v)$.

A tree T satisfies *an XPath expression p if there is a node v such that $T \models p(v_0, v)$, where v_0 is the root node of T.*

Definition 6. *An XPath expression p is* satisfiable *under a DTD D if some $T \in TL(D)$ satisfies p. Moreover, an XPath expression p is* valid *under a DTD D if every $T \in TL(D)$ satisfies p.*

2.3 XML Schema Mappings

In this section, we define XML schema mappings according to [1,4,5].

Definition 7. *A* dependency *is an expression of the form* $\pi \longrightarrow \pi'$*, where* π *and* π' *are XPath expressions called* document patterns*. An* XML schema mapping $\mathcal{M}$ *is a triple* (D_S, D_T, Σ)*, where* D_S *is a DTD representing a source schema,* D_T *is a DTD representing a target schema, and* Σ *is a finite set of dependencies. Hereafter, XML schema mappings are referred to as just mappings.*

Originally, in [1,4,5], document patterns are defined by means of tree patterns. However, we use XPath expressions in order to exploit many results on XPath satisfiability and validity.

Definition 8. *A pair of trees* $T_S \in TL(D_S)$ *and* $T_T \in TL(D_T)$ satisfies *a dependency* $\pi \longrightarrow \pi'$ *if the following condition is satisfied: If* T_S *satisfies* π*, then* T_T *satisfies* π'*.*

The set of all pairs of trees which satisfy all dependencies of $\mathcal{M}$ is denoted as $[\![\mathcal{M}]\!]$. Then, consistency and absolute consistency are defined as follows:

Definition 9. *Let* $\mathcal{M} = (D_S, D_T, \Sigma)$*.* $\mathcal{M}$ *is* consistent *if* $[\![\mathcal{M}]\!] \neq \emptyset$*. On the other hand,* $\mathcal{M}$ *is* absolutely consistent *if, for every tree* T_S *conforming to* D_S*, there exists a tree* T_T *conforming to* D_T *such that* $(T_S, T_T) \in [\![\mathcal{M}]\!]$*.*

3 Deciding Consistency

3.1 Deciding Consistency of Mappings between General DTDs

To begin with, we show that the consistency decision problem can be solved by deciding validity and satisfiability of combinations of document patterns in dependencies. The following lemma is essentially equivalent to the statement given in the proof of the EXPTIME-completeness of deciding consistency under general DTDs [5].

Lemma 1. *Let* $\mathcal{M} = (D_S, D_T, \Sigma)$ *and* $\Sigma = \{\pi_i \longrightarrow \pi'_i \mid i \in [n]\}$ *where* $[n] = \{1, 2, \ldots, n\}$*. Then,* $\mathcal{M}$ *is consistent if and only if there are two disjoint subsets* $I = \{i_1, \cdots, i_k\}$ *and* $J = \{j_1, \cdots, j_{k'}\}$ *of* $[n]$ *such that* $I \cup J = [n]$*,*

- $\cdot :: r_S[\pi_{i_1} \vee \cdots \vee \pi_{i_k}]$ *is not valid under* D_S*, and*
- $\cdot :: r_T[\pi'_{j_1} \wedge \cdots \wedge \pi'_{j_{k'}}]$ *is satisfiable under* D_T

where r_S *and* r_T *are the root labels of* D_S *and* D_T*, respectively.*

Proof. Suppose $\mathcal{M}$ is consistent. Then, a pair of trees (T_S, T_T) belongs to $[\![\mathcal{M}]\!]$. Let $J = \{j_1, \cdots, j_{k'}\}$ be a subset of $[n]$ such that T_T satisfies π'_{j_l} for each $j_l \in J$. Then, because of the semantics of qualifier $[\,]$, T_T satisfies $\cdot :: r_T[\pi'_{j_1} \wedge \cdots \wedge \pi'_{j_{k'}}]$. On the other hand, let $I = \{i_1, \cdots, i_k\} = [n] - J$. Then, T_T does not satisfy π'_i for any $i \in I$. So, because $(T_S, T_T) \in [\![\mathcal{M}]\!]$, T_S must not satisfy π_i for any $i \in I$. Therefore, because of the semantics of $\vee$, T_S does not satisfy $\cdot :: r_S[\pi_{i_1} \vee \cdots \vee \pi_{i_k}]$.

Conversely, suppose that $\cdot :: r_S[\pi_{i_1} \vee \cdots \vee \pi_{i_k}]$ is not valid under D_S and $\cdot :: r_T[\pi'_{j_1} \wedge \cdots \wedge \pi'_{j_{k'}}]$ is satisfiable under D_T for some $I = \{i_1, \cdots, i_k\}$ and $J = \{j_1, \cdots, j_{k'}\}$ such that $I \cap J = \emptyset$ and $I \cup J = [n]$. The non-validity of $\cdot :: r_S[\pi_{i_1} \vee \cdots \vee \pi_{i_k}]$ under D_S means that there is an instance $T_S \in TL(D_S)$ which does not satisfy π_i for any $i \in I$. On the other hand, if $\cdot :: r_T[\pi'_{j_1} \wedge \cdots \wedge \pi'_{j_{k'}}]$ is satisfiable under D_T, there is $T_T \in TL(D_T)$ which satisfies π'_j for all $j \in J = [n] - I$. Therefore, the pair (T_S, T_T) belongs to $[\![\mathcal{M}]\!]$ because T_S does not satisfy π_l or T_T satisfies π'_l for each $\pi_l \longrightarrow \pi'_l \in \Sigma$ where $l \in [n]$. □

According to Lemma 1, the problem of deciding consistency of schema mappings can be solved by using the decision procedure of validity and satisfiability of XPath expressions under DTDs. However, the complexity of this decision procedure for consistency of schema mappings between general DTDs is high, because deciding validity and satisfiability of XPath expressions under general DTDs is known to be intractable [6] and combinatorial explosion occurs in the choice of disjoint subsets I and J of $[n]$. In fact, consistency of mappings between general DTDs is known to be EXPTIME-complete [5] even if the class of document patterns is restricted to downward axes and qualifiers.

3.2 Deciding Consistency of Mappings between DC-DTDs

Now we restrict source and target schemas to be DC-DTDs. We shall give tractability results of consistency of mappings between DC-DTDs by showing the following facts:

1. Satisfiability for broad XPath subclasses under DC-DTDs is decidable in polynomial time [2,3].
2. Validity for XPath expressions in $\mathcal{X}$ (i.e., the whole XPath class of this paper) under DC-DTDs is decidable in polynomial time.
3. Under DC-DTDs, we can decide consistency of a schema mapping by deciding validity or satisfiability of each individual document pattern in a mapping instead of combinations of document patterns.

First, we recall that satisfiability under DC-DTDs for several XPath subclasses can be decided in polynomial time [2,3]. Table 3 shows the known tractability results of XPath satisfiability under DC-DTDs.

Next, we show that validity of XPath expressions in $\mathcal{X}$ under DC-DTDs is tractable. For a DC regular expression $e = e_1 e_2 \cdots e_n$, let $\tilde{e}$ denote the regular expression obtained by replacing with ϵ each subexpression e_i such that $e_i = (e'_i)^*$ for some regular expression e'_i . For a DC-DTD $D = (\Gamma, r, P)$, let $\tilde{D} = (\Gamma, r, \tilde{P})$ such that $\tilde{P}(l) = \tilde{e}$ when $P(l) = e$. Note that $TL(\tilde{D})$ is a singleton because each content model $\tilde{e}$ in $\tilde{D}$ does not contain any disjunction operator and thus it represents only one sequence over Γ. The DTD $\tilde{D}$ has the following property on validity of XPath expressions under a DC-DTD D.

Lemma 2. *For an XPath expression π in $\mathcal{X}$ and a DC-DTD D, the unique tree $\tilde{t}$ in $TL(\tilde{D})$ satisfies π if and only if π is valid under D.*

Table 3. Computational complexity of the satisfiability decision under DC-DTDs

$\downarrow$	$\downarrow^*$	$\uparrow$	$\uparrow^*$	$\rightarrow^+$	$\leftarrow^+$	$\cup$	[ ]	Satisfiability
+	+			+	+	+	+	PTIME
+	+	+	+	+	+	+		PTIME
+		+		+			+	PTIME
+		+	+				+	NP-complete
+		+				+	+	NP-complete
	+		+				+	NP-complete
+		+		+	+		+	NP-complete

Proof. For the if part, assume that π is valid under D. Then, since $\tilde{t} \in TL(D)$, $\tilde{t}$ satisfies π. For the only if part, assume that $\tilde{t}$ satisfies π. From the definition of $\tilde{D}$, for each $l \in \Gamma$, the unique sequence in $L(\tilde{P}(l))$ is a subsequence of any $w \in L(P(l))$. Let t be an arbitrary tree in $TL(D)$. Then there is an injective mapping θ from the set of nodes of $\tilde{t}$ to that of t such that

- $\lambda(v) = \lambda(\theta(v))$ for each node v of $\tilde{t}$,
- for any two nodes v_1 and v_2 of $\tilde{t}$, if v_1 is a parent of v_2, then $\theta(v_1)$ is a parent of $\theta(v_2)$, and
- for any two nodes v_1 and v_2 of $\tilde{t}$, if v_1 is a following sibling of v_2, then $\theta(v_1)$ is a following sibling of $\theta(v_2)$.

For this, we can see by induction on the structure of π that $\tilde{t} \models \pi(v, v')$ implies $t \models \pi(\theta(v), \theta(v'))$. Thus, t also satisfies π. □

Lemma 2 leads to tractability of XPath validity checking under DC-DTDs. That is, validity of an XPath expression π in $\mathcal{X}$ under a DC-DTD D is linear-time reducible to evaluation of π on the tree $\tilde{t} \in TL(\tilde{D})$. Since a polynomial-time algorithm for evaluation is known [7], validity of XPath expressions in $\mathcal{X}$ under DC-DTDs can be decided in polynomial time.

Lastly, we can avoid combinatorial explosion mentioned in Section 3.1 by the following lemmas.

Lemma 3. *Let π_1 and π_2 be XPath expressions in $\mathcal{X}$ and D be a DC-DTD. Either or both of π_1 and π_2 is valid under D if and only if $\cdot :: r[\pi_1 \vee \pi_2]$ is valid under D, where r is the root label of D.*

Proof. The only if part is trivial. For the if part, assume that $\cdot :: r[\pi_1 \vee \pi_2]$ is valid under D. Consider $\tilde{t} \in TL(\tilde{D})$. Because $\tilde{t} \in TL(D)$, $\tilde{t}$ satisfies $\cdot :: r[\pi_1 \vee \pi_2]$. By the semantics of $\vee$, $\tilde{t}$ satisfies at least one of π_1 and π_2. Without loss of generality, we assume that $\tilde{t}$ satisfies π_1. From Lemma 2, π_1 is valid under D. □

Lemma 4. *Let π_1 and π_2 be XPath expressions in $\mathcal{X}$ and D be a DC-DTD. Both π_1 and π_2 are satisfiable under D if and only if $\cdot :: r[\pi_1 \wedge \pi_2]$ is satisfiable under D, where r is the root label of D.*

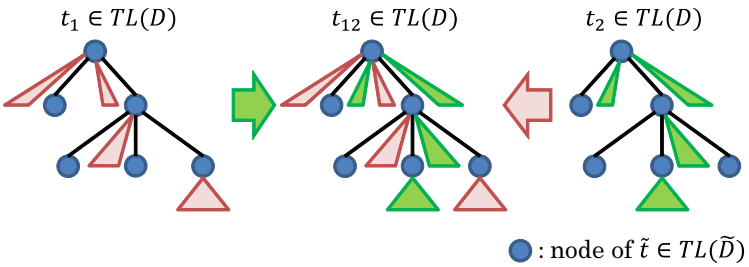

Fig. 1. Construction of t_{12} from t_1 and t_2 by inserting subtrees

Proof. The if part is trivial. For the only if part, assume that both π_1 and π_2 are satisfiable under D. Then, there are trees t_1 and t_2 in $TL(D)$ such that t_1 and t_2 satisfy π_1 and π_2, respectively. Because $\mathcal{X}$ does not include negation operator and next-sibling axis, if a tree t satisfies XPath expression $\pi \in \mathcal{X}$, any tree obtained by inserting arbitrary subtrees to t also satisfies π [3]. Thus, if we can obtain the same tree t_{12} from t_1 and t_2 by inserting subtrees (see Fig. 1), t_{12} satisfies $\cdot :: r[\pi_1 \wedge \pi_2]$. To show that such t_{12} exists, we should prove that for any regular DC expression e, if w_1, $w_2 \in L(e)$, there is $w \in L(e)$ such that w_1 and w_2 are subsequences of w. This property holds because DC regular expressions cannot specify non-co-occurrence among labels. □

From the above, we can refine Lemma 1 for deciding consistency of mappings between DC-DTDs as follows.

Lemma 5. *Let $\mathcal{M}$ be a mapping (D_S, D_T, Σ) such that the schemas D_S and D_T are DC-DTDs. Then, $\mathcal{M}$ is consistent if and only if there is no dependency $\pi \longrightarrow \pi'$ in Σ such that π is valid under D_S and π' is not satisfiable under D_T.*

Proof. Let $\Sigma = \{\pi_i \longrightarrow \pi'_i \mid i \in [n]\}$ and r_S and r_T be the root labels of D_S and D_T, respectively. From Lemma 1, we prove this lemma by showing the next condition: there is no $\pi \longrightarrow \pi' \in \Sigma$ such that π is valid under D_S and π' is not satisfiable under D_T if and only if there are two disjoint subsets $I = \{i_1, \ldots, i_k\}$ and $J = \{j_1, \ldots, j_{k'}\}$ of $[n]$ such that $I \cup J = [n]$, $\cdot :: r_S[\pi_{i_1} \vee \cdots \vee \pi_{i_k}]$ is not valid under D_S, and $\cdot :: r_T[\pi'_{j_1} \wedge \cdots \wedge \pi'_{j_{k'}}]$ is satisfiable under D_T.

Assume that there is no $\pi \longrightarrow \pi' \in \Sigma$ such that π is valid under D_S and π' is not satisfiable under D_T. Then, for each $l \in [n]$, π_l is not valid under D_S or π'_l is satisfiable under D_T. Let $I = \{i_1, \ldots, i_k\}$ be the set of all the indexes i_l in $[n]$ such that π_{i_l} is not valid under D_S. Let $J = \{j_1, \ldots, j_{k'}\} = [n] - I$. Then π'_{j_l} is satisfiable under D_T for each $j_l \in J$. From Lemmas 3 and 4, $\cdot :: r_S[\pi_{i_1} \vee \cdots \vee \pi_{i_k}]$ is not valid under D_S and $\cdot :: r_T[\pi'_{j_1} \wedge \cdots \wedge \pi'_{j_{k'}}]$ is satisfiable under D_T.

Conversely, assume that there are two disjoint subsets $I = \{i_1, \ldots, i_k\}$ and $J = \{j_1, \ldots, j_{k'}\}$ of $[n]$ such that $I \cup J = [n]$, $\cdot :: r_S[\pi_{i_1} \vee \cdots \vee \pi_{i_k}]$ is not valid under D_S, and $\cdot :: r_T[\pi'_{j_1} \wedge \cdots \wedge \pi'_{j_{k'}}]$ is satisfiable under D_T. From Lemmas 3 and 4, π_l is not valid under D_S for any $l \in I$ and π'_l is satisfiable under D_T for any $l \in J$. Since $I \cup J = [n]$, there is no $\pi \longrightarrow \pi' \in \Sigma$ such that π is valid under D_S and π' is not satisfiable under D_T. □

Here, in order to decide consistency of mappings between DC-DTDs, we just have to check validity of π under D_S and satisfiability of π' under D_T for each $\pi \longrightarrow \pi'$ in Σ. Therefore, if we choose as the class of document patterns the XPath subclasses in which validity and satisfiability under DC-DTDs are tractable, deciding consistency of schema mappings is tractable.

Theorem 1. *Let $\mathcal{M}$ be a mapping (D_S, D_T, Σ), where D_S, D_T are DC-DTDs, such that for each $\pi \longrightarrow \pi' \in \Sigma$, $\pi \in \mathcal{X}$ and $\pi' \in \mathcal{X}_T$, where $\mathcal{X}_T$ is an XPath subclass for which satisfiability is decidable in polynomial time under DC-DTD (see Table 3). Then, consistency of $\mathcal{M}$ can be decided in polynomial time.*

4 Deciding Absolute Consistency

4.1 Deciding Absolute Consistency of Mappings between General DTDs

We begin with a lemma stating that absolute consistency of a mapping $\mathcal{M} = (D_S, D_T, \Sigma)$ can be determined by checking satisfiability of all XPath expressions induced by all subset of Σ. Again, this lemma is essentially equivalent to the statement in the explanation of Π_2^p-completeness of absolute consistency of mappings between general DTDs [1].

Lemma 6. *Mapping $\mathcal{M} = (D_S, D_T, \Sigma)$ is absolutely consistent if and only if the following condition holds: For every subset $\Sigma' = \{\pi_1 \longrightarrow \pi'_1, \ldots, \pi_k \longrightarrow \pi'_k\}$ of Σ, if $\cdot :: r_S[\pi_1 \wedge \cdots \wedge \pi_k]$ is satisfiable under D_S, then $\cdot :: r_T[\pi'_1 \wedge \cdots \wedge \pi'_k]$ is satisfiable under D_T, where r_S and r_T are the root labels of D_S and D_T, respectively.*

Proof. Suppose that mapping $\mathcal{M} = (D_S, D_T, \Sigma)$ is absolutely consistent, and consider an arbitrary subset $\Sigma' = \{\pi_1 \longrightarrow \pi'_1, \ldots, \pi_k \longrightarrow \pi'_k\}$ of Σ. Also suppose that $T_S \in TL(D_S)$ satisfies $\cdot :: r_S[\pi_1 \wedge \cdots \wedge \pi_k]$. Since $\mathcal{M}$ is absolutely consistent, there is $T_T \in TL(D_T)$ such that $(T_S, T_T) \in [\![\mathcal{M}]\!]$. Hence, T_T satisfies $\cdot :: r_T[\pi'_1 \wedge \cdots \wedge \pi'_k]$ and the only if part holds.

Conversely, suppose that for every subset $\Sigma' = \{\pi_1 \longrightarrow \pi'_1, \ldots, \pi_k \longrightarrow \pi'_k\}$ of Σ, if $\cdot :: r_S[\pi_1 \wedge \cdots \wedge \pi_k]$ is satisfiable under D_S, then $\cdot :: r_T[\pi'_1 \wedge \cdots \wedge \pi'_k]$ is satisfiable under D_T. Consider an arbitrary tree $T_S \in TL(D_S)$. Let $\Sigma' = \{\pi_1 \longrightarrow \pi'_1, \ldots, \pi_k \longrightarrow \pi'_k\}$ be a subset of Σ such that T_S satisfies π_i for each i $(1 \leq i \leq k)$. Then, T_S satisfies $\cdot :: r_S[\pi_1 \wedge \cdots \wedge \pi_k]$, and therefore, $\cdot :: r_S[\pi_1 \wedge \cdots \wedge \pi_k]$ is satisfiable under D_S. Then, by assumption, there must be $T_T \in TL(D_T)$ such that T_T satisfies $\cdot :: r_T[\pi'_1 \wedge \cdots \wedge \pi'_k]$, and hence, $(T_S, T_T) \in [\![\mathcal{M}]\!]$. Consequently, $\mathcal{M}$ is absolutely consistent. □

Similarly to the case of consistency, the above lemma tells us that the hardness of deciding absolute consistency stems from the combinatorial explosion caused by checking all the subset of dependencies as well as the hardness of deciding XPath satisfiability.

4.2 Deciding Absolute Consistency of Mappings between DC-DTDs

To obtain tractability results on absolute consistency, we focus on DC-DTDs again. Similarly to the case of consistency, the combinatorial explosion of subsets of dependencies can be avoided, that is, satisfiability can be independently checked for each dependency.

Lemma 7. *Let D_S and D_T be DC-DTDs. Mapping $\mathcal{M} = (D_S, D_T, \Sigma)$ is absolutely consistent if and only if for every dependency $\pi \longrightarrow \pi' \in \Sigma$, if π is satisfiable under D_S, then π' is satisfiable under D_T.*

Proof. Suppose that mapping $\mathcal{M} = (D_S, D_T, \Sigma)$ is absolutely consistent. Then, by Lemma 6, for every subset $\Sigma' = \{\pi_1 \longrightarrow \pi'_1, \ldots, \pi_k \longrightarrow \pi'_k\}$ of Σ, if $\cdot :: r_S[\pi_1 \wedge \cdots \wedge \pi_k]$ is satisfiable under D_S, then $\cdot :: r_T[\pi'_1 \wedge \cdots \wedge \pi'_k]$ is satisfiable under D_T. This immediately implies the only if part of the lemma.

Conversely, suppose that for every dependency $\pi \longrightarrow \pi' \in \Sigma$, if π is satisfiable under D_S, then π' is satisfiable under D_T. By the property of DC-DTDs (Lemma 4), two XPath expressions π_1 and π_2 are satisfiable under DC-DTD D if and only if $\cdot :: r[\pi_1 \wedge \pi_2]$ is satisfiable under D, where r is the root label of D. Hence, for any subset $\Sigma' = \{\pi_1 \longrightarrow \pi'_1, \ldots, \pi_k \longrightarrow \pi'_k\}$ of Σ, $\pi_1, \ldots, \pi_k$ are satisfiable under D_S if and only if $\cdot :: r_S[\pi_1 \wedge \cdots \wedge \pi_k]$ is satisfiable under D_S. On the other hand, by the assumption, if $\pi_1, \ldots, \pi_k$ are satisfiable under D_S, then $\pi'_1, \ldots, \pi'_k$ are satisfiable under D_T, and hence, $\cdot :: r_T[\pi'_1 \wedge \cdots \wedge \pi'_k]$ is satisfiable under D_T. Therefore, the if part holds. □

Now, the next theorem immediately follows from the above lemma:

Theorem 2. *Let $\mathcal{M}$ be a mapping (D_S, D_T, Σ), where D_S, D_T are DC-DTDs, such that for each $\pi \longrightarrow \pi' \in \Sigma$, $\pi, \pi' \in \mathcal{X}_T$, where $\mathcal{X}_T$ is an XPath subclass for which satisfiability is decidable in polynomial time under DC-DTD (see Table 3). Then, absolute consistency of $\mathcal{M}$ can be decided in polynomial time.*

5 Conclusion

This paper has discussed the consistency and absolute consistency problems of XML schema mappings between DC-DTDs. First, we have shown that consistency of mappings between DC-DTDs can be solved by deciding validity and satisfiability of XPath expressions linearly many times. Moreover, we have proved that the XPath validity problem in the presence of DC-DTDs can be solved in polynomial time. From these facts, we have proved that the consistency of mappings between DC-DTDs can be solved in polynomial time if the document patterns are in an XPath class for which satisfiability and validity are tractable under DC-DTDs. Next, we have shown that absolute consistency of the mappings between DC-DTDs can also be solved by deciding satisfiability of XPath expressions in the presence of DC-DTD linearly many times. So the absolute consistency of mappings between DC-DTDs can also be solved in polynomial time if the document patterns are in an XPath class for which satisfiability is tractable under DC-DTDs.

As stated in Section 1, there is a superclass of DC-DTDs, called $DC^{?+}$-DTDs [3]. They allow two operators ? (zero or one occurrence) and + (one or more occurrences) of regular expressions in a restricted manner. Actually, $DC^{?+}$-DTDs are also a proper superclass of nested-relational DTDs. The tractability results of absolute consistency of mappings between DC-DTDs can be extended to mappings between $DC^{?+}$-DTDs because $DC^{?+}$-DTDs inherit all the tractability of XPath satisfiability of DC-DTDs. However, as for consistency, the tractability results do not seem to be extended to mappings between $DC^{?+}$-DTDs because we are conjecturing that XPath validity under $DC^{?+}$-DTDs is coNP-hard. Now we are trying to prove the coNP-hardness and then propose a DTD class which is a superclass of both nested-relational DTDs and DC-DTDs but under which XPath validity is tractable.

We are also planning to incorporate data values into mappings. As stated in [1], consistency of mappings with data values can be reduced to consistency of mappings without data values. However, such reduction is not possible for absolute consistency. It is interesting to find XPath classes such that absolute consistency of mappings between DC-DTDs with data values becomes tractable.

Acknowledgment. The authors thank Mr. Yohei Kusunoki of Osaka University for his insightful comments and suggestions on deciding XPath validity under DC-DTDs. The authors are also thankful to the anonymous reviewers for their helpful comments. This research is supported in part by Grant-in-Aid for Scientific Research (C) 23500120 from Japan Society for the Promotion of Science.

References

1. Amano, S., Libkin, L., Murlak, F.: XML schema mappings. In: Proceedings of the Twenty-Eigth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 33–42 (2009)
2. Ishihara, Y., Morimoto, T., Shimizu, S., Hashimoto, K., Fujiwara, T.: A tractable subclass of DTDs for XPath satisfiability with sibling axes. In: Gardner, P., Geerts, F. (eds.) DBPL 2009. LNCS, vol. 5708, pp. 68–83. Springer, Heidelberg (2009)
3. Ishihara, Y., Shimizu, S., Fujiwara, T.: Extending the tractability results on XPath satisfiability with sibling axes. In: Lee, M.L., Yu, J.X., Bellahsène, Z., Unland, R. (eds.) XSym 2010. LNCS, vol. 6309, pp. 33–47. Springer, Heidelberg (2010)
4. Arenas, M., Libkin, L.: XML data exchange: Consistency and query answering. Journal of the ACM 55(2) (2008)
5. Arenas, M., Barcelo, P., Libkin, L., Murlak, F.: Relational and XML Data Exchange. Morgan & Claypool (2010)
6. Benedikt, M., Fan, W., Geerts, F.: XPath satisfiability in the presence of DTDs. Journal of the ACM 55(2) (2008)
7. Bojańczyk, M., Parys, P.: XPath evaluation in linear time. Journal of the ACM 58(4), 17 (2011)