XPath Satisfiability with Parent Axes or Qualifiers Is Tractable under Many of Real-World DTDs

Yasunori Ishihara (Osaka University)
Nobutaka Suzuki (University of Tsukuba)
Kenji Hashimoto (NAIST)
Shogo Shimizu (Gakushuin Women's College)
Toru Fujiwara (Osaka University)

XPath satisfiability

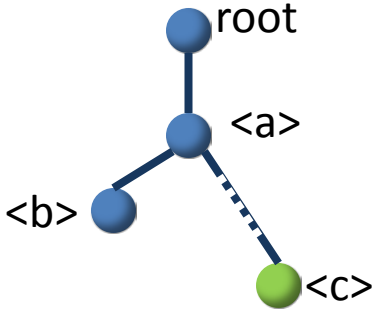
- Input: XPath expression p
  DTD D
- Output: Is there an XML document T such that
  - T conforms to D and
  - p returns a nonempty set for T?

- Research on XPath satisfiability is motivated by query optimization
  - Unsatisfiable (parts of) XPath expressions can be replaced with the empty set

XPath expression

- Atomic expression: "*axis*::*label*"
  - $\downarrow$ (child axis)
  - $\downarrow^*$ (descendant-or-self axis)
  - $\uparrow$ (parent axis)
  - $\uparrow^*$ (ancestor-or-self axis)
  - $\rightarrow^+$ (following-sibling axis)
  - $\leftarrow^+$ (preceding-sibling axis)
- Path constructors:
  - / (path concatenation)
  - $\cup$ (path union)
  - [ ] (qualifier, possibly with $\wedge$ and $\vee$)

Ordinary notation: /a[b]//c



Our notation: ($\downarrow$::a[$\downarrow$::b])/$\downarrow^*$::c

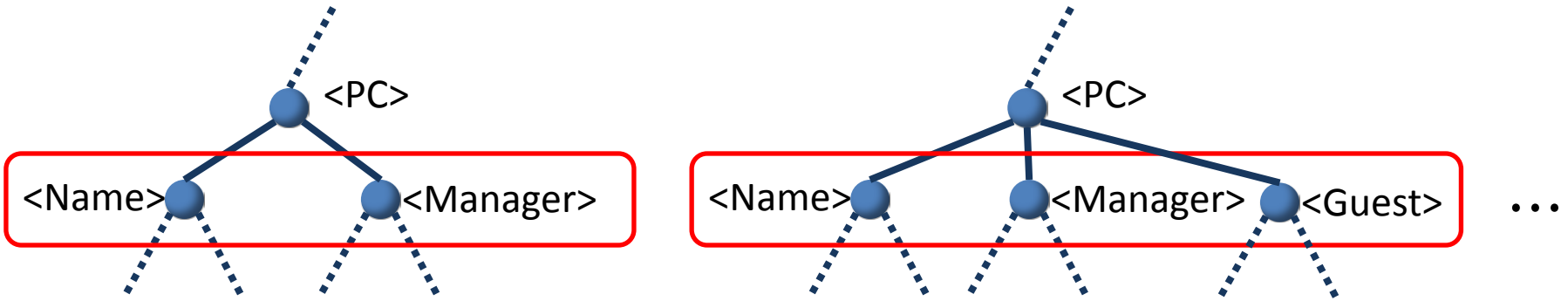
No negation operators

Document type definition (DTD)

- A DTD
  - specifies a set of XML documents
  - is naturally modeled by a tree grammar
    - Each production rule specifies, for a label, a set of sequences of its children by a regular expression

PC -> Name Manager (Manager | Guest)*

content model



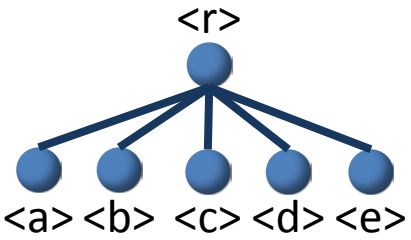
Difficulty in XPath satisfiability

- XPath satisfiability under *arbitrary DTDs* is in P for a very small subclass of XPath [BFG05,BFG08,GF05]
  - $(\downarrow, \downarrow^*, \cup)$
- Analyzing *non-cooccurrence of sibling labels* is difficult
  - non-cooccurrence is specified by *disjunctions*

XPath exp.: $\downarrow^*$::r [↓::a] [↓::b] [↓::c] [↓::d] [↓::e]

DTD: r -> (ad | be)(b | ace)(ae | cd)

x_1 $\bar{x}_1$ x_2 $\bar{x}_2$ x_3 $\bar{x}_3$



$$\varphi = \underbrace{(x_1 \vee \bar{x}_2 \vee x_3)}_{a} \wedge \underbrace{(\bar{x}_1 \vee x_2)}_{b} \wedge \underbrace{(\bar{x}_2 \vee \bar{x}_3)}_{c} \wedge \underbrace{(x_1 \vee \bar{x}_3)}_{d} \wedge \underbrace{(\bar{x}_1 \vee \bar{x}_2 \vee x_3)}_{e}$$

Related work & our purpose

- Two approaches:
  - Tackling the intractability of XPath satisfiability itself [GL06,GL07,GLS07]
    - XPath expressions and DTDs are translated into formulas in monadic second-order (MSO) logic or in a variant of μ-calculus
    - Satisfiability is verified by fast decision procedures for MSO or μ-calculus formulas
  - Finding subclasses of DTDs such that satisfiability of a larger XPath class becomes tractable

DTD classes restricting disjunctions (1)

- Disjunction-free DTD [BFG05,BFG08,GF05]
  - No content model contains disjunction operators of regular expressions
    - non-cooccurrence of lables cannot be specified
  - Tractable XPath classes [IMSHF09]:
    - $(\downarrow, \downarrow^*, \rightarrow^+, \leftarrow^+, \cup, [\ ])$
    - $(\downarrow, \downarrow^*, \uparrow, \uparrow^*, \rightarrow^+, \leftarrow^+, \cup)$
  - Disjunction-freeness is too restrictive from the practical point of view

DTD classes restricting disjunctions (2)

- Disjunction-capsuled DTD (DC-DTD) [IMSHF09], $DC^{?+\#}$-DTD [IHSF12]

$a\#b = a \mid b \mid ab$

  - Regular expression operators: $\cdot$, | , $*$, ?, +, #
  - Every disjunction is in the scope of $*$ or +
    - non-cooccurrence cannot be specified

PC -> Name$^{?}$ Manager (Manager | Guest)*

PC -> (Name | IP)$^{?}$ Manager (Manager | Guest)*

  - disjunction-free $\subset$ DC $\subset$ $DC^{?+\#}$
  - All tractability results of disjunction-free DTDs are inherited by $DC^{?+\#}$-DTDs
    - as long as the XPath class is within our formulation

DTD class restricting non-coocurrence

- Duplicate-free DTD (DF-DTD) [MWM07]
  - Regular expression operators: $\cdot$, | , $*$, ?, +
  - Each label <u>appears at most once</u> in a content model
    - Non-cooccurrence of sibling labels exists but can be easily analyzed

PC -> (Name | IP)(Manager | Guest)*

PC -> (Name | IP) Manager (Manager | Guest)*

  - Tractable XPath classes:
    - $(\downarrow, [\,]_\wedge)$ [MWM07]
    - $(\downarrow, \uparrow, \rightarrow^+, \leftarrow^+)$ [SF09]

$[\,]_\wedge$: qualifier with only $\wedge$

Hybridizing DF-DTDs and $DC^{?+\#}$-DTDs

- RW-DTDs [IHSF12]

  - 26 out of 27 real-world DTDs are RW-DTDs
  - 1406 out of 1407 real-world DTD rules are covered
  - Expected that RW-DTDs has the same tractability as DF-DTDs
    - only DF parts can specify non-cooccurrence

Hybridizing the two DTD classes

- RW-DTDs [IHSF12]

  - 26 out of 27 real-world DTDs are RW-DTDs
  - 1406 out of 1407 real-world DTD rules are covered

↓	↓*	↑	↑*	→+	←+	∪	[ ]∧	[ ]	any	RW	DF	DC?+#
+	+					+			P	P	P	P
+	+			+	+				NPC	P	P	P
+		+							NPC	NPC	P	P
+							+		NPC	NPC	P	P

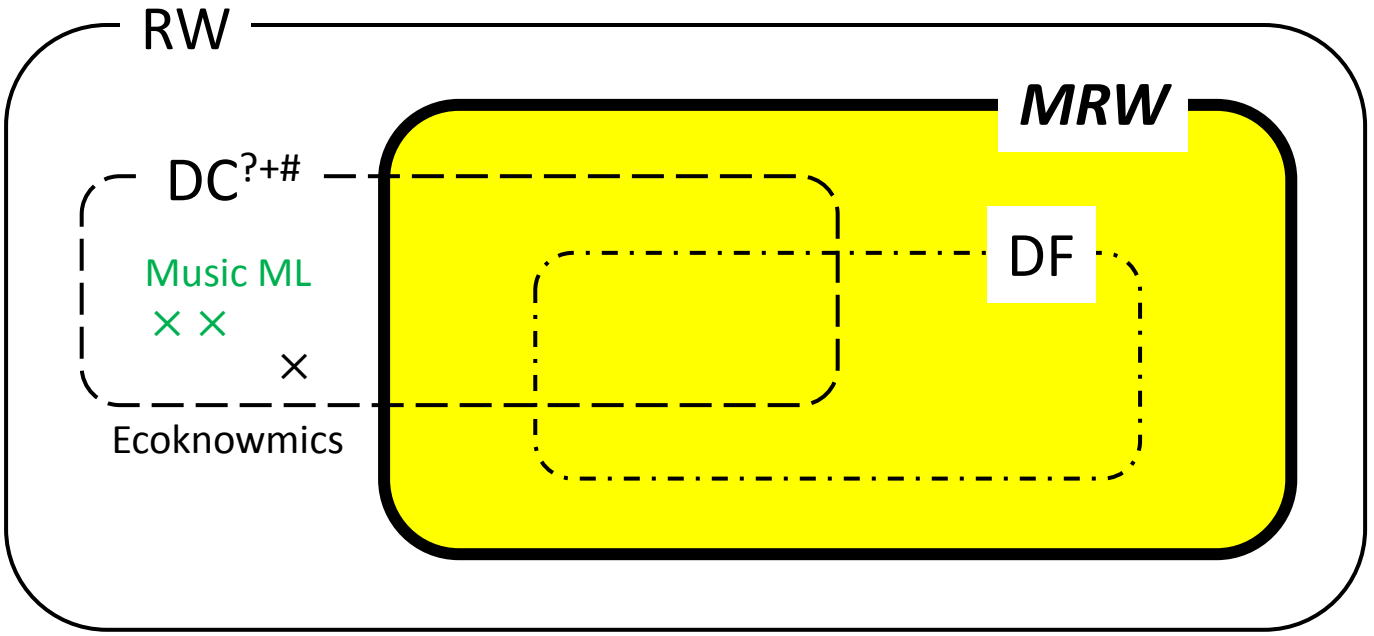
[ ]∧: qualifier with only ∧

gap

NPC: NP-complete

Contribution of this work

- *MRW-DTDs*:
  - 24 out of 27 real-world DTDs are MRW-DTDs
  - 1403 out of 1407 real-world DTD rules are covered



Contribution of this work

- *MRW-DTDs*:
  - 24 out of 27 real-world DTDs are MRW-DTDs
  - 1403 out of 1407 real-world DTD rules are covered

↓	↓*	↑	↑*	→+	←+	∪	[ ]∧	[ ]	RW	**MRW**	DF	DC?+#
+	+			+	+				P	P	P	P
+		+		+	+				NPC	**P**	P	P
+				+	+		+		NPC	**P**	**P**	P
+						+	+	+	NPC	NPC	NPC	P
	+						+		NPC	NPC	NPC	P
+	+	+							NPC	NPC	NPC	P

[ ]∧: qualifier with only ∧

NPC: NP-complete

Outline

- Results on RW-DTDs [IHSF12]
- MRW-DTDs and their tractability results
- Conclusion

RW-DTDs [IHSF12]

- Hybridization of DF-DTDs and DC$^{?+\#}$-DTDs

  - 26 out of 27 real-world DTDs are RW-DTDs
  - 1406 out of 1407 real-world DTD rules are covered

↓	↓*	↑	↑*	→+	←+	∪	[ ]∧	[ ]	any	RW	DF	DC?+#
+	+					+			P	P	P	P
+	+			+	+				NPC	P	P	P
+		+							NPC	NPC	P	P
+							+		NPC	NPC	P	P

[ ]∧: qualifier with only ∧

gap

NPC: NP-complete

Satisfiability checking algorithm for $(\downarrow, \downarrow^*, \rightarrow^+, \leftarrow^+)$ under RW-DTDs

1. DTD transformation

PC -> (Name | IP) Manager (Manager | Guest)*

PC -> Name · IP · Manager (Manager | Guest)*

RW

$DC^{?+\#}$

2. Approximate satisfiability checking
   - Run the known, efficient algorithm for $DC^{?+\#}$-DTDs
   - The algorithm may answer "satisfiable" mistakenly
3. Consistency checking
   - Check whether p is consistent with the non-cooccurrence of labels specified by the original RW-DTD
   - p is unsatisfiable if p says "Name and IP are siblings"

Difficulty for $(\downarrow, \uparrow)$ and $(\downarrow, [\,]_\wedge)$ under RW-DTDs

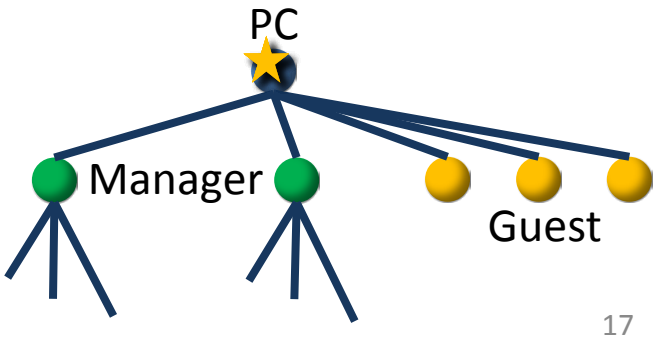
- Label occurrence of some bounded, plural number of times

PC -> (Name | IP) Manager · Manager · Guest*

  - $(\downarrow, \downarrow^*, \rightarrow^+, \leftarrow^+)$: goes down only
  - $(\downarrow, \uparrow)$, $(\downarrow, [\,]_\wedge)$: goes down and up many times

p: checks Manager's children many times

At consistency checking step,
we have to decide nondeterministically:
``Which Manager should we go to?''



Outline

- Results on RW-DTDs [IHSF12]
- MRW-DTDs and their tractability results
- Conclusion

MRW-DTDs

- RW-DTDs with the following restriction:
  - label a is outside the scope of any * and +
    ⇒ label a appears only once in the content model

PC -> (Name | IP) Manager · Guest*

PC -> (Name | IP) Manager · Manager · Guest*

PC -> (Name | IP) Manager^{+} (Manager | Guest)*

PC -> (Name | IP) Manager (Manager | Guest)*

- Each label appears “at most once” or “unboundedly many times” in a content model

Satisfiability checking algorithm under MRW-DTDs

1. DTD transformation (MRW -> $DC^{?+\#}$)
2. Approximate satisfiability checking
3. Consistency checking
   - Check if p is consistent with the non-cooccurrence of sibling labels specified by the original MRW-DTD

– Maintain sibling information of all the nodes that may be revisited during the traverse by p

– MRW-DTDs:

always revisited

always avoidable to be revisited

- Each label appears “at most once” or “unboundedly many times” in a content model

Satisfiability check for $(\downarrow, \uparrow, \rightarrow^+, \leftarrow^+)$

- Always-revisited nodes:
  - nodes with labels outside the scope of any * and +
  - ancestor nodes of the current node
    - due to ↑
- XPath expressions are non-branching
    - due to absence of $[\,]_\wedge$

XPath

$$((\downarrow :: r/\rightarrow^+ :: b)/(\downarrow :: a/\uparrow :: b))/\rightarrow^+ :: c$$

sibling information

DTD

$$r \rightarrow r^* (a^* b \mid c) r^*$$
$$b \rightarrow a$$

Satisfiability check for $(\downarrow, \uparrow, \rightarrow^+, \leftarrow^+)$

- Always-revisited nodes:
  - nodes with labels outside the scope of any * and +
  - ancestor nodes of the current node
    - due to ↑
- XPath expressions are non-branching
    - due to absence of $[\,]_\wedge$

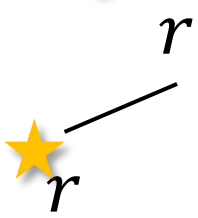
XPath

$$((\downarrow :: r/\rightarrow^+ :: b)/(\downarrow :: a/\uparrow :: b))/\rightarrow^+ :: c$$

DTD

$$r \rightarrow r^*\,(a^*\,b \mid c)\,r^*$$
$$b \rightarrow a$$

sibling information



Satisfiability check for $(\downarrow, \uparrow, \rightarrow^+, \leftarrow^+)$

- Always-revisited nodes:
  - nodes with labels outside the scope of any * and +
  - ancestor nodes of the current node
    - due to ↑
- XPath expressions are non-branching
  - due to absence of $[\,]_\wedge$

XPath

$$((\downarrow :: r/\rightarrow^+ :: b)/(\downarrow :: a/\uparrow :: b))/\rightarrow^+ :: c$$

DTD

$$r \rightarrow r^*\,(a^*\,b \mid c)\,r^*$$
$$b \rightarrow a$$

sibling information

Satisfiability check for $(\downarrow, \uparrow, \rightarrow^+, \leftarrow^+)$

- Always-revisited nodes:
  - nodes with labels outside the scope of any * and +
  - ancestor nodes of the current node
    - due to ↑
- XPath expressions are non-branching
    - due to absence of $[\,]_\wedge$

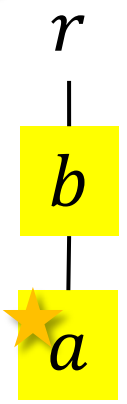
XPath

$$((\downarrow :: r/\rightarrow^+ :: b)/(\downarrow :: a/\uparrow :: b))/\rightarrow^+ :: c$$

DTD

$$r \rightarrow r^*\,(a^*\,b \mid c)\,r^*$$
$$b \rightarrow a$$

sibling information



Satisfiability check for $(\downarrow, \uparrow, \rightarrow^+, \leftarrow^+)$

- Always-revisited nodes:
  - nodes with labels outside the scope of any * and +
  - ancestor nodes of the current node
    - due to $\uparrow$
- XPath expressions are non-branching
    - due to absence of $[\,]_\wedge$

XPath

$$((\downarrow :: r/\rightarrow^+ :: b)/(\downarrow :: a/\uparrow :: b))/\rightarrow^+ :: c$$

DTD

$$r \rightarrow r^*\,(a^*\,b \mid c)\,r^*$$
$$b \rightarrow a$$

sibling information

Satisfiability check for $(\downarrow, \uparrow, \rightarrow^+, \leftarrow^+)$

- Always-revisited nodes:
  - nodes with labels outside the scope of any * and +
  - ancestor nodes of the current node
    - due to $\uparrow$
- XPath expressions are non-branching
    - due to absence of $[\,]_\wedge$

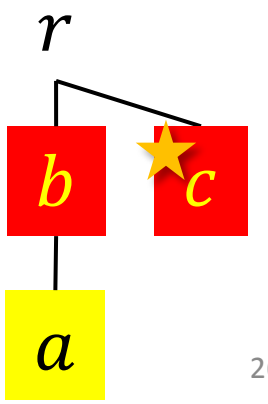
XPath

$$((\downarrow :: r/\rightarrow^+ :: b)/(\downarrow :: a/\uparrow :: b))/\rightarrow^+ :: c$$

DTD

$$r \rightarrow r^*\,(a^*\,b \mid c)\,r^*$$
$$b \rightarrow a$$

sibling information

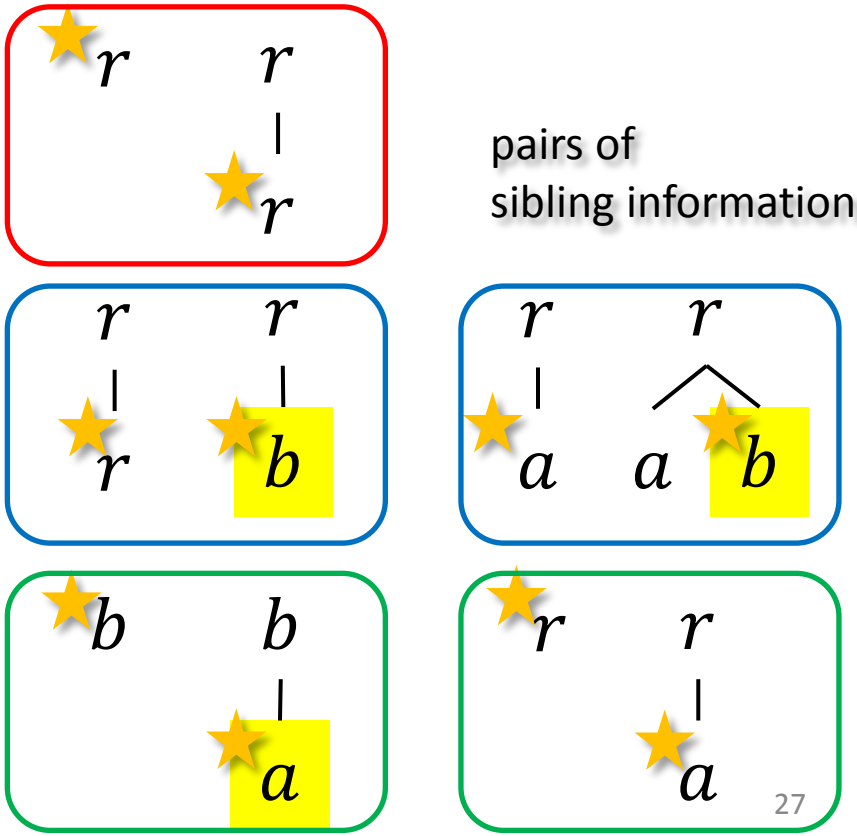


Satisfiability check for $(\downarrow, \rightarrow^+, \leftarrow^+, [\,]_\wedge)$

- Always-revisited nodes:
  - nodes with labels outside the scope of any * and +
    - due to absence of ↑
- XPath expressions are branching
  - due to $[\,]_\wedge$

XPath $\downarrow :: r / \rightarrow^+ :: b[\downarrow :: a]$

DTD
$$r \rightarrow r^* (a^* b \mid c) r^*$$
$$b \rightarrow a$$

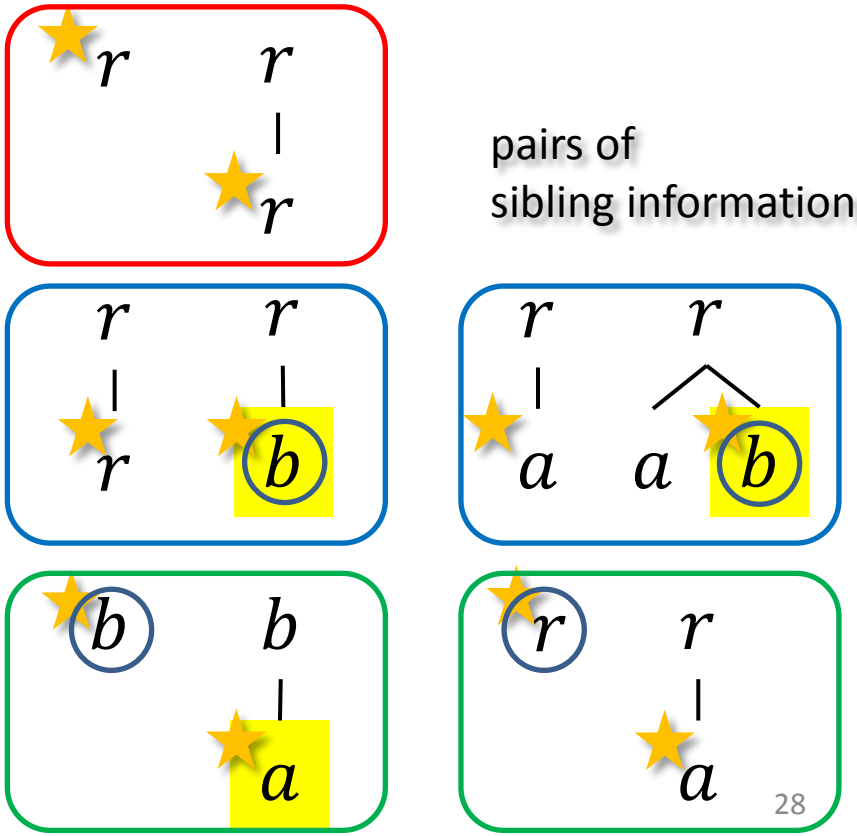


Satisfiability check for $(\downarrow, \rightarrow^+, \leftarrow^+, [\,]_\wedge)$

- Always-revisited nodes:
  - nodes with labels outside the scope of any * and +
    - due to absence of ↑
- XPath expressions are branching
  - due to $[\,]_\wedge$

XPath $\downarrow :: r / \rightarrow^+ :: b[\downarrow :: a]$

DTD
$$r \rightarrow r^* (a^* b \mid c) r^*$$
$$b \rightarrow a$$



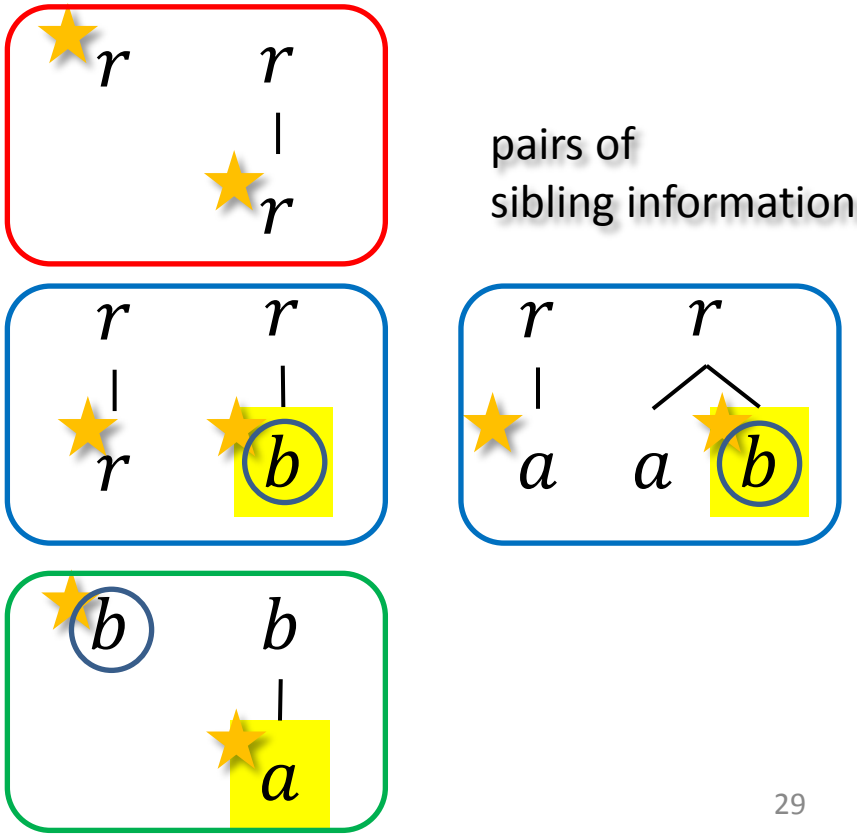
pairs of sibling information

Satisfiability check for $(\downarrow, \rightarrow^+, \leftarrow^+, [\,]_\wedge)$

- Always-revisited nodes:
  - nodes with labels outside the scope of any * and +
    - due to absence of ↑
- XPath expressions are branching
    - due to $[\,]_\wedge$

XPath $\downarrow :: r / \rightarrow^+ :: b[\downarrow :: a]$

DTD
$$r \rightarrow r^* (a^* b \mid c) r^*$$
$$b \rightarrow a$$

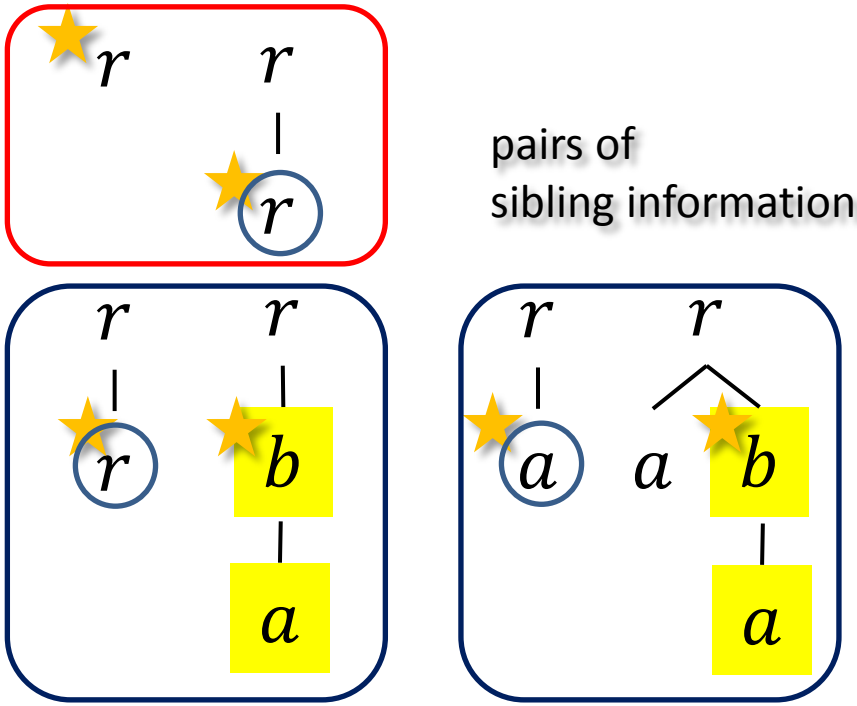


Satisfiability check for $(\downarrow, \rightarrow^+, \leftarrow^+, [\,]_\wedge)$

- Always-revisited nodes:
  - nodes with labels outside the scope of any * and +
    - due to absence of ↑
- XPath expressions are branching
  - due to $[\,]_\wedge$

XPath $\downarrow :: r / \rightarrow^+ :: b[\downarrow :: a]$

DTD
$$r \rightarrow r^* (a^* b \mid c) r^*$$
$$b \rightarrow a$$



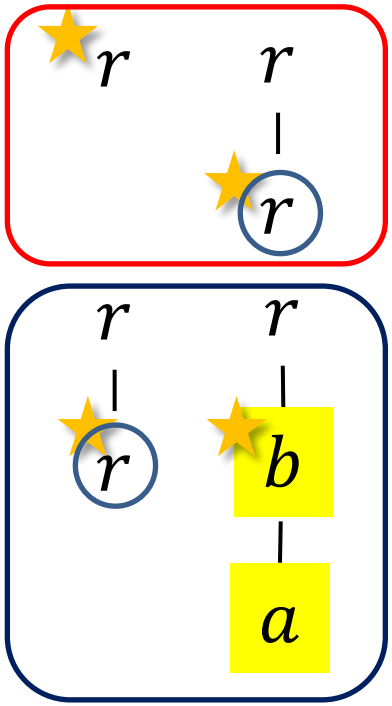
pairs of sibling information

Satisfiability check for $(\downarrow, \rightarrow^+, \leftarrow^+, [\,]_\wedge)$

- Always-revisited nodes:
  - nodes with labels outside the scope of any * and +
    - due to absence of ↑
- XPath expressions are branching
    - due to $[\,]_\wedge$

XPath $\downarrow :: r / \rightarrow^+ :: b[\downarrow :: a]$

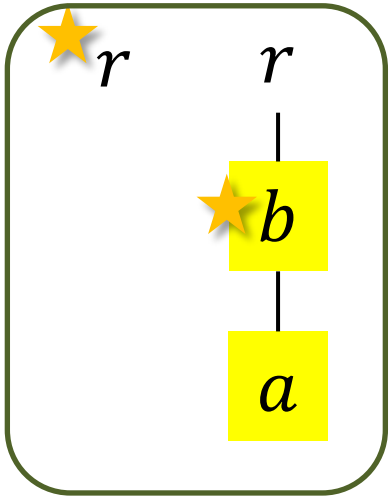
DTD
$$r \rightarrow r^* (a^* b \mid c) r^*$$
$$b \rightarrow a$$



pairs of sibling information

Satisfiability check for $(\downarrow, \rightarrow^+, \leftarrow^+, [\,]_\wedge)$

- Always-revisited nodes:
  - nodes with labels outside the scope of any * and +
    - due to absence of ↑
- XPath expressions are branching
    - due to $[\,]_\wedge$



pairs of sibling information

XPath $\downarrow :: r/\rightarrow^+ :: b[\downarrow :: a]$

DTD $r \rightarrow r^*\,(a^*\,b \mid c)\,r^*$

$b \rightarrow a$

More formal discussion

- *Schema graph* G of a given MRW-DTD D:
  - A directed graph representing the topology of D
- *Satisfaction* of p by G
- Theorem:

  $\exists T \exists w \exists w',\ T \vDash p(w, w')$

  - T: tree conforming to D
  - w, w': node sequences of T from the root

  iff

  $\exists \theta \exists \beta \exists \beta',\ G \vDash p((\theta(w), \beta), (\theta(w'), \beta'))$

  - θ: correspondence between the nodes of T and G
  - β, β': sibling information

Conclusion

- *MRW-DTDs*:
  - 24 out of 27 real-world DTDs are MRW-DTDs
  - 1403 out of 1407 real-world DTD rules are covered

↓	↓*	↑	↑*	→+	←+	∪	[ ]∧	[ ]	RW	**MRW**	DF	DC?+#
+	+			+	+				P	P	P	P
+		+		+	+				NPC	**P**	P	P
+				+	+		+		NPC	**P**	**P**	P
+						+	+	+	NPC	NPC	NPC	P
	+						+		NPC	NPC	NPC	P
+	+	+							NPC	NPC	NPC	P

[ ]∧: qualifier with only ∧

NPC: NP-complete

Future work

- Complexity for $(\downarrow, \uparrow, \rightarrow^{+}, \leftarrow^{+}, [\,]_{\wedge})$ under MRW-DTDs
  - Reduction from 3SAT seems difficult
  - Merging two strategies of satisfiability check also seems difficult

- Comparison with the other approach
  - which uses fast decision procedures for MSO and μ-calculus formulas

References

- [BFG05] Benedikt, M., Fan, W., Geerts, F.: XPath satisfiability in the presence of DTDs. In: Proceedings of the Twenty-fourth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. (2005) 25-36
- [BFG08] Benedikt, M., Fan, W., Geerts, F.: XPath satisfiability in the presence of DTDs. Journal of the ACM 55(2) (2008) 1-79
- [GF05] Geerts, F., Fan, W.: Satisfiability of XPath queries with sibling axes. In: Proceedings of the 10th International Symposium on Database Programming Languages. (2005) 122-137
- [GL06] Genevès, P., Layaïda, N.: A system for the static analysis of XPath. ACM Transactions on Information Systems 24(4) (2006) 475-502
- [GL07] Genevès, P., Layaïda, N.: Deciding XPath containment with MSO. Data & Knowledge Engineering 63(1) (2007) 108-136
- [GLS07] Genevès, P., Layaïda, N., Schmitt, A.: Efficient static analysis of XML paths and types. In: Proceedings of the ACM SIGPLAN 2007 Conference on Programming Language Design and Implementation. (2007) 342-351
- [IHSF12] Ishihara, Y., Hashimoto, K., Shimizu, S., Fujiwara, T.: XPath satisfiability with downward and sibling axes is tractable under most of real-world DTDs. In: Proceedings of the 12th International Workshop on Web Information and Data Management. (2012) 11-18
- [IMSHF09] Ishihara, Y., Morimoto, T., Shimizu, S., Hashimoto, K., Fujiwara, T.: A tractable subclass of DTDs for XPath satisfiability with sibling axes. In: Proceedings of the 12th International Symposium on Database Programming Languages. (2009) 68-83
- [ISF10] Ishihara, Y., Shimizu, S., Fujiwara, T.: Extending the tractability results on XPath satisfiability with sibling axes. In: Proceedings of the 7th International XML Database Symposium. (2010) 33-47
- [MWM07] Montazerian, M., Wood, P.T., Mousavi, S.R.: XPath query satisfiability is in PTIME for real-world DTDs. In: Proceedings of the 5th International XML Database Symposium, LNCS 4704. (2007) 17-30
- [SF09] Suzuki, N., Fukushima, Y.: Satisfiability of simple XPath fragments in the presence of DTD. In: Proceedings of the 11th International Workshop on Web Information and Data Management. (2009) 15-22