Journal of Applied Logic 8 (2010) 153–172

Contents lists available at ScienceDirect

Journal of Applied Logic

www.elsevier.com/locate/jal

Complete axiomatizations for XPath fragments

Balder ten Cate [a], Tadeusz Litak [b,*], Maarten Marx [c]

[a] INRIA, ENS de Cachan, 61, avenue du President Wilson, 94235 Cachan Cedex, France
[b] Department of Computer Science and Information Systems, Birkbeck College, University of London, Malet Street, Bloomsbury, London, WC1E 7HX, UK
[c] Informatics Institute, University of Amsterdam, Kruislaan 403, 1098 SJ, Amsterdam, The Netherlands

ARTICLE INFO

Article history:
Available online 22 September 2009

Keywords:
XML
XPath
Query rewriting
Axiomatization
Modal logic

ABSTRACT

We provide complete axiomatizations for several fragments of Core XPath, the navigational core of XPath 1.0 introduced by Gottlob, Koch and Pichler. A complete axiomatization for a given fragment is a set of equivalences from which every other valid equivalence is derivable; equivalences can be thought of as (undirected) rewrite rules. Specifically, we axiomatize single axis fragments of Core XPath as well as full Core XPath. Our completeness proofs use results and techniques from modal logic.

XPath [36] is the W3C standard language for navigating through XML documents. It lies at the core of most XML processing technologies, such as XML Schema, XQuery and XSLT. Because of the central rôle that XPath plays in many XML-related formalisms, the static analysis of XPath expressions has been a prominent subject of research. The complexity of containment and satisfiability has been determined for many fragments of XPath (cf. [29] and references therein). Query optimization plays a key part in most XML processing engines, and is typically implemented by means of rewrite rules (cf. for example [1,18]).

In this paper, we present complete axiomatizations for fragments of XPath. By a complete axiomatization, we mean a set of valid equivalence schemes between XPath expressions, from which every other valid equivalence is derivable. An example of a valid equivalence scheme is $A[\phi \vee \psi] \equiv A[\phi] \cup A[\psi]$. No matter which path expression is substituted for A and which node expressions are substituted for ϕ and ψ, the resulting pair of path expressions will be equivalent.

From the point of view of query optimization, the equivalence schemes can be thought of as undirected rewrite rules. Completeness, then, means that whenever an expression Γ is equivalent to another expression Δ, then Δ can be obtained from Γ by a sequence of applications of these rewrite rules. Thus, a complete axiomatization provides a good starting point for obtaining an effective set of rewrite rules. Of course, having a complete axiomatization is not enough for query optimization: one also needs a good rewriting strategy. This is another topic, which we will not pursue here.

In this paper, we study Core XPath, which was introduced in [8,9] to capture the navigational core of XPath 1.0. We present finite complete axiomatizations for all *single axis fragments* of Core XPath. The completeness holds both for node expressions and for path expressions. We also obtain complete axiomatizations for node expressions and path expressions of the full Core XPath language. The latter axiomatization is *non-orthodox*, i.e., requires introducing an additional inference rule breaking the chain of equational reasoning. We leave the existence of a finite orthodox complete axiomatization for path expressions of Core XPath as an open problem.

Related work and the meaning of our results. We are aware of two complete axiomatizations for XPath fragments. The first is for the downward, positive, filter-free fragment of XPath [2], a rather limited fragment, and the second [31] concerns the

* Corresponding author.
E-mail addresses: balder.tencate@uva.nl (B. ten Cate), tadeusz@dcs.bbk.ac.uk (T. Litak), maartenmarx@uva.nl (M. Marx).

1570-8683/$ – see front matter
doi:10.1016/j.jal.2009.09.002

powerful language of Core XPath 2.0. However, an axiomatization for Core XPath is of interest even though an axiomatization for Core XPath 2.0 is already known. If one wants to use the latter to prove validity of Core XPath (1.0) equivalences, one may need to use intermediate Core XPath 2.0 expressions in the derivation. It is not clear, or even likely, that each of these intermediate expressions could be rewritten to an equivalent one in Core XPath. This is an important issue, as Core XPath 2.0 is a much more complex language than Core XPath: its query equivalence problem is non-elementary [29], while the same problem for Core XPath is only ExpTime-complete [4,19].

It is notable in this respect that for the axiomatization for Core XPath presented here, if two equivalent expressions both belong to the same single axis fragment, a derivation exists that uses only intermediate results *from the same fragment*. Observe that the complexity of the equivalence problem for single axis fragments of Core XPath is again typically lower than for full Core XPath: it is either NP-complete or PSpace-complete, depending on the axis, as can be derived from results in modal logic (cf. [28]).

Many typical XPath expressions belong to single axis fragments of Core XPath. Here are some examples.

Core XPath ($\downarrow$)	`chapter/*[not(subsection)]`
Core XPath ($\downarrow^+$)	`//footnote`
Core XPath ($\uparrow$)	`../..[not(..)]`
Core XPath ($\leftarrow^+$)	`preceding-sibling::price`
Core XPath ($\rightarrow^+$)	`following-sibling::modification[not(following-sibling::modification)]`

Thus, it is interesting to learn that the transformation between two equivalent expression in the same fragment can be always conducted—at least in principle—using only the most basic rules of equational reasoning and without any intermediate expressions from more complex XPath fragments, where combinations of transitive/non-transitive, forward- and backward-looking or horizontal and vertical axes would be allowed.

We briefly comment on logical meaning of our results. Our paper transfers equational axiomatization results for "static" modal logics of node expressions to the "dynamic" or *procedural* [32] setting of path expressions. That is, we obtain axiomatizations for corresponding *modal transition logics* [35]. This can be done in an uniform way if we allow non-equational, non-orthodox inference rules such as *Sep* discussed in Section 4. The task, however, becomes non-trivial if we insist that the dynamic ("procedural") axiomatization is also purely equational. Even if the original static logic is in the *basic modal signature* of a single unary operator (in the terminology of this paper, it corresponds to simple node expressions of some *single axis fragment*), smooth transfer of completeness for the corresponding modal transition logic requires a good choice of proof technique. Our work seems to confirm again the importance of *normal modal forms* [6,15,23]. Sections 2.1 and 3.2 below offer brief comments and references on algebraic structures corresponding to the signature of Core XPath.

1. Preliminaries

1.1. Semantics: XML trees

We abstract away from atomic data attached to the individual elements, and view an XML document as a finite node-labelled sibling-ordered tree. More formally, given a countably infinite set $\mathsf{lab} = v_1, v_2, v_3, \ldots$ of node labels, we define an *XML tree* to be a structure $T = (N, R_\downarrow, R_\rightarrow, L)$, where

- $(N, R_\downarrow)$ is a finite tree (with N the set of nodes and $R_\downarrow$ the child relation),
- $R_\rightarrow$ is the successor relation of some linear ordering between siblings in the tree (in particular, for $R_{\rightarrow^+}$ the transitive closure of $R_\rightarrow$, we have that for any two distinct siblings x, y, either $xR_{\rightarrow^+}y$ or $yR_{\rightarrow^+}x$), and
- $L : N \to \wp(\mathsf{lab})$ labels the nodes with elements of lab.

We denote by $R_{\downarrow^+}$ and $R_{\rightarrow^+}$ the transitive closures of $R_\downarrow$ and $R_\rightarrow$, i.e., the *descendant* and *following-sibling* relations. $R_\leftarrow$, $R_{\leftarrow^+}$, $R_\uparrow$ and $R_{\uparrow^+}$ are the converses of, respectively, $R_\rightarrow$, $R_{\rightarrow^+}$, $R_\downarrow$ and $R_{\downarrow^+}$. The elements of lab correspond to XML tags. It is customary to require that each node satisfies precisely one tag. For technical reasons it is convenient for us not to make this assumption from the start, but we will explain later how a suitable axiom schema capturing this can be added to the axiomatization.

1.2. Core XPath, the navigational core of XPath 1.0

The Core XPath language was introduced in [8,9] in order to capture the navigational core of XPath 1.0. Here, we follow the definition of Core XPath given in [29], which has a different notation and is slightly more expressive, due to the fact that it includes the non-transitive sibling axes.

The syntax of Core XPath is defined as follows:

$$\mathsf{Step} := \downarrow \mid \leftarrow \mid \uparrow \mid \rightarrow$$

$$\mathsf{Axis} := \mathsf{Step} \mid \mathsf{Step}^+$$

Table 1
Comparison with unabbreviated XPath notation [36].

Ours vs.	Unabbreviated notation	Ours vs.	Unabbreviated notation
$\downarrow$	`child::*`	$\uparrow$	`parent::*`
$\leftarrow$	`preceding-sibling::*[position()=1]`	$\rightarrow$	`following-sibling::*[position()=1]`
$\downarrow^+$	`descendant::*`	$\uparrow^+$	`ancestor::*`
$\leftarrow^+$	`preceding-sibling::*`	$\rightarrow^+$	`following-sibling::*`
v	`self::`v	$\langle$pexpr$\rangle$	pexpr
$\neg$nexpr	`not`(nexpr)	nexpr $\vee$ nexpr	nexpr `or` nexpr

Table 2
Meaning of Core XPath expressions in $(N, R_\downarrow, R_\rightarrow, L)$.

$[\![.]\!]_{\mathsf{PExpr}}$	$:=$	$\{(x,x) \mid x \in N\}$
$[\![\mathsf{s}]\!]_{\mathsf{PExpr}}$	$:=$	R_{s}
$[\![\mathsf{s}^+]\!]_{\mathsf{PExpr}}$	$:=$	the transitive closure of R_{s}
$[\![A/B]\!]_{\mathsf{PExpr}}$	$:=$	$\{(x,y) \mid \exists z$ such that $(x,z) \in [\![A]\!]_{\mathsf{PExpr}}$ and $(z,y) \in [\![B]\!]_{\mathsf{PExpr}}\}$
$[\![A \cup B]\!]_{\mathsf{PExpr}}$	$:=$	$[\![A]\!]_{\mathsf{PExpr}} \cup [\![B]\!]_{\mathsf{PExpr}}$
$[\![A[\phi]]\!]_{\mathsf{PExpr}}$	$:=$	$\{(x,y) \mid (x,y) \in [\![A]\!]_{\mathsf{PExpr}}$ and $y \in [\![\phi]\!]_{\mathsf{NExpr}}\}$
$[\![v]\!]_{\mathsf{NExpr}}$	$:=$	$\{x \mid v \in L(x)\}$
$[\![\langle\mathsf{PathEx}\rangle]\!]_{\mathsf{NExpr}}$	$:=$	$\{x \mid \exists y.(x,y) \in [\![\mathsf{PathEx}]\!]_{\mathsf{PExpr}}\}$
$[\![\neg\phi]\!]_{\mathsf{NExpr}}$	$:=$	$\{x \mid x \notin [\![\phi]\!]_{\mathsf{NExpr}}\}$
$[\![\phi_1 \vee \phi_2]\!]_{\mathsf{NExpr}}$	$:=$	$[\![\phi_1]\!]_{\mathsf{NExpr}} \cup [\![\phi_2]\!]_{\mathsf{NExpr}}$

$$\mathsf{PathEx} := . \mid \mathsf{Axis} \mid \mathsf{PathEx}[\mathsf{NodeEx}] \mid \mathsf{PathEx}/\mathsf{PathEx} \mid \mathsf{PathEx} \cup \mathsf{PathEx}$$

$$\mathsf{NodeEx} := v \mid \langle\mathsf{PathEx}\rangle \mid \neg\mathsf{NodeEx} \mid \mathsf{NodeEx} \vee \mathsf{NodeEx} \ (v \in \mathsf{lab})$$

In this paper, Greek letters $\phi, \psi, \ldots$ range over elements of NodeEx and Roman capitals $A, B, C, \ldots$ over elements of PathEx. The letters s and a are metavariables ranging over, respectively, elements of Step and elements of Axis; thus, we can write $\mathsf{a} := \mathsf{s} \mid \mathsf{s}^+$. It is customary in XPath to call "." an axis, but for our purposes it is convenient to treat it as a separate constant. Note that we include the non-transitive sibling axes $\rightarrow$ and $\leftarrow$ in the language. Also, we use angled brackets to distinguish path expressions from node expressions that test for the existence of a path. We use the following abbreviations:

true	for the node expression $\langle.\rangle$	$\perp$	for the path expression .[false]
false	for the node expression $\neg$true	$\phi \wedge \psi$	for the node expression $\neg(\neg\phi \vee \neg\psi)$

The reader familiar with original *unabbreviated XPath notation* [36] will notice that we included a number of abbreviations and alterations—see Table 1 for comparison. Official *abbreviated XPath notation* [36] is different both from the unabbreviated one and from the one in the present paper.

The semantics of Core XPath is defined in Table 2 by the functions $[\![\cdot]\!]_{\mathsf{PExpr}}$ and $[\![\cdot]\!]_{\mathsf{NExpr}}$ which take as input an XML tree and a path expression or node expression, and produce a binary relation over the set of nodes or a set of nodes, respectively. The XML tree is kept implicit in our notation.

For arbitrary $\mathsf{A} \subseteq \mathsf{Axis}$, we will denote by Core XPath(A) the fragment of Core XPath in which the only allowed axes are those listed in A. When A has only one element a, we call Core XPath(a) a *single axis fragment of* Core XPath.

A Core XPath *node equivalence instance* is an expression of the form $\phi \equiv \psi$, where $\phi, \psi \in \mathsf{NodeEx}$, and a Core XPath *path equivalence instance* is an expression of the form $A \equiv B$, where $A, B \in \mathsf{PathEx}$. An *equivalence scheme* may also contain *path metavariables* (capital Latin letters) and *node metavariables* (lowercase Greek letters) as subexpressions of either side of the equivalence. All expressions in Table 3 are equivalence schemes. An *instance of a scheme* is any equivalence instance arising from the given scheme by uniformly replacing all metavariables with concrete expressions. For example, $\langle.[\langle\downarrow^+\rangle]\rangle \equiv \langle\downarrow^+\rangle$ is one of infinitely many instances of NdAx2 in Table 3. Depending on the context, we use the notion *equivalence* to refer both to equivalence schemes and equivalence instances whenever it does not lead to ambiguities. Also, when equivalence schemes are used in an axiomatization, we may refer to them as *axioms*. An a-*instance* of an axiom scheme is an instance where no axis constant other than a occurs (recall that "." is not an axis in our terminology!).

An equivalence is a Core XPath(A) *equivalence* if the expressions on both sides belong to Core XPath(A). If in every XML tree it holds that $[\![\phi]\!]_{\mathsf{NExpr}} = [\![\psi]\!]_{\mathsf{NExpr}}$, respectively $[\![A]\!]_{\mathsf{PExpr}} = [\![B]\!]_{\mathsf{PExpr}}$, then we say that the equivalence instance is *valid*. An equivalence scheme is valid if all of its instances are.

We use $A \leqslant B$ as shorthand for $A \cup B \equiv B$, and $\phi \leqslant \psi$ as shorthand for $\phi \vee \psi \equiv \psi$. While this approach to containment is standard in logic and algebra, it may at times prove confusing for the database community. This definition still allows to derive equivalence through pairwise containment whenever necessary—it is enough to show that expressions on both sides are equivalent to their join and use transitivity of equivalence. However, containment defined this way can be used to reason

Table 3
Basic axiom schemes valid on arbitrary structures.

Path Axiom Schemes for idempotent semirings			
ISAx1	$(A \cup B) \cup C$	$\equiv$	$A \cup (B \cup C)$
ISAx2	$A \cup B$	$\equiv$	$B \cup A$
ISAx3	$A \cup A$	$\equiv$	A
ISAx4	$A/(B/C)$	$\equiv$	$(A/B)/C$
ISAx5	$./A$	$\equiv$	A
	$A/.$	$\equiv$	A
ISAx6	$A/(B \cup C)$	$\equiv$	$A/B \cup A/C$
	$(A \cup B)/C$	$\equiv$	$A/C \cup B/C$
ISAx7	$\bot \cup A$	$\equiv$	A
Path Axiom Schemes for predicates			
PrAx1	$A[\neg\langle B\rangle]/B$	$\equiv$	$\bot$
PrAx2	$.[\langle . \rangle]$	$\equiv$	$.$
PrAx3	$A[\phi \vee \psi]$	$\equiv$	$A[\phi] \cup A[\psi]$
PrAx4	$(A/B)[\phi]$	$\equiv$	$A/B[\phi]$
Node Axiom Schemes			
NdAx1	ϕ	$\equiv$	$\neg(\neg\phi \vee \psi) \vee \neg(\neg\phi \vee \neg\psi)$
NdAx2	$\langle .[\phi]\rangle$	$\equiv$	ϕ
NdAx3	$\langle A \cup B\rangle$	$\equiv$	$\langle A\rangle \vee \langle B\rangle$
NdAx4	$\langle A/B\rangle$	$\equiv$	$\langle A[\langle B\rangle]\rangle$

about equivalence even when it holds only in one direction: it abbreviates equivalence between the greater expression and the join of both. The reason why containment is not always handled this way in the database world is that the most studied class of queries—i.e., conjunctive queries—is not closed under union, whereas all classes of expressions studied in this paper are.

A note of caution. The reader should be aware that some valid equivalence schemes for Core XPath are not valid equivalence schemes for the full XPath language. For example, $A[\phi][\psi] = A[\phi \wedge \psi]$ is a valid equivalence of Core XPath, but not of XPath, as is witnessed by the following instance:

$$\texttt{child} :: *[\texttt{child} :: v][\texttt{position}() = 1] \not\equiv \texttt{child} :: *[\texttt{child} :: v \;\texttt{and}\; \texttt{position}() = 1]$$

Note, though, that $\texttt{position}() = 1$ is not a Core XPath node expression despite the fact that some expressions containing it can be simulated—see clauses for $\leftarrow$ and $\rightarrow$ in Table 1.

2. Axioms, derivations and node normal forms

Recall that an axiomatization is a finite set of axioms—valid equivalence schemes. While it is not obligatory that all axioms contain metavariables (see TransAx2 in Table 4), most axioms do and thus are schemes in proper sense. Tables 3, 4 and 5 present our proposed axioms for single axis fragments and (node expressions of) full Core XPath. The rest of the paper will be devoted to proofs of their completeness. In this section, we are going to explain them in more detail, discuss the notion of derivation we are going to use and prove a preliminary result on normal form for node expressions, which is going to be used in all proofs to follow.

2.1. Basic axioms

Table 3 presents basic axiom schemes for Core XPath, which do not involve any axis or axes. As we are going to see, however, it is possible to prove a completeness result for one of the single axis fragments even on the basis of this minimal set of axioms—namely for Core XPath($\downarrow$).

2.1.1. Idempotent semirings axioms

The name comes from algebra. Idempotency is the property expressed by the axiom ISAx3. The natural numbers with addition and multiplication form a semiring, but not an idempotent one. Distributive lattices with the two basic connectives *meet* and *join* are natural examples of idempotent semirings. *Tarski's relation algebras* [26,27] and *Kleene algebras* [16,17] interpret / and $\cup$ in the same way as we do (as composition and union of relations, respectively), hence both have idempotent semirings reducts.

Table 4
Axiom schemes for single axis fragments.

All axiom schemes from Table 3: common for all axes				
Additional Axiom Schemes for children axis				
none				
Additional Axiom Scheme for other nontransitive axes				
LinNTAx	$\mathsf{s}[\neg\phi]$	$\equiv$	$.[\neg\langle \mathsf{s}[\phi]\rangle]/\mathsf{s}$	for $\mathsf{s} \in \{\rightarrow, \leftarrow, \uparrow\}$
Additional Axiom Schemes for all transitive axes				
TransAx1	$\langle \mathsf{s}^+[\phi]\rangle$	$\equiv$	$\langle \mathsf{s}^+[\phi \wedge \neg\langle \mathsf{s}^+[\phi]\rangle]\rangle$	for $\mathsf{s} \in \{\rightarrow, \leftarrow, \uparrow, \downarrow\}$
TransAx2	$\mathsf{s}^+/\mathsf{s}^+$	$\leqslant$	s^+	
Additional Axiom Scheme for non-descendant transitive axes				
TransAx3	$.[\langle \mathsf{s}^+[\phi]\rangle]/\mathsf{s}^+$	$\equiv$	$\mathsf{s}^+[\phi] \cup \mathsf{s}^+[\phi]/\mathsf{s}^+ \cup \mathsf{s}^+[\langle \mathsf{s}^+[\phi]\rangle]$	for $\mathsf{s} \in \{\rightarrow, \leftarrow, \uparrow\}$

Table 5
Axiom schemes for full Core XPath.

All axiom schemes from Table 3 plus				
TransAx1	$\langle \mathsf{s}^+[\phi]\rangle$	$\equiv$	$\langle \mathsf{s}^+[\phi \wedge \neg\langle \mathsf{s}^+[\phi]\rangle]\rangle$	for $\mathsf{s} \in \{\rightarrow, \leftarrow, \uparrow, \downarrow\}$
TreeAx1	$\mathsf{s}^+/\mathsf{s} \cup \mathsf{s}$	$\equiv$	s^+	for $\mathsf{s} \in \{\rightarrow, \leftarrow, \uparrow, \downarrow\}$
	$\mathsf{s}/\mathsf{s}^+ \cup \mathsf{s}$	$\equiv$	s^+	
TreeAx2	$\mathsf{s}[\phi]/\mathsf{s}^{-1}$	$\equiv$	$.[\langle \mathsf{s}[\phi]\rangle]$	for $\mathsf{s} \in \{\leftarrow, \rightarrow, \downarrow\}$
TreeAx3	$\uparrow[\phi]/\downarrow$	$\equiv$	$(\leftarrow^+ \cup \rightarrow^+ \cup .)[\langle \uparrow[\phi]\rangle]$	
TreeAx4	$\leftarrow^+$	$\equiv$	$\leftarrow^+[\langle\uparrow\rangle]$	
	$\rightarrow^+$	$\equiv$	$\rightarrow^+[\langle\uparrow\rangle]$	

2.1.2. Predicate axioms

In the signature of Core XPath 2.0 [31], which includes intersection and complementation operators for binary relations, predicates can be treated as defined operations—just like in the one-sorted signature of Tarski's relation algebras [26,27]. In XPath 1.0, there are less operations on relations available and predicates cannot be defined away.

2.1.3. Node axioms

NdAx1 is known in the algebraic community as the *Huntington equation* [14,13,22] and together with Der1 and Der2 from Table 6 allows to derive the axioms of Boolean algebra for the node expression connectives $\neg$ and $\vee$. The axioms NdAx2, NdAx3 and NdAx4 are counterparts of PrAx2, PrAx3 and PrAx4, respectively. This slight redundancy is the price one pays for working in a two-sorted signature.

An algebraic note on the two-sorted signature. As pointed out by Georg Struth (p.c.), the two-sorted setting we are working with corresponds to reducts of structures known to algebraists as *boolean modules* (see [11] for an extensive discussion and up-to-date references). Strictly speaking, our signature is contained properly between that of *semiring modules* [5] and that of boolean modules; the most appropriate name here would seem to be *antidomain semiring modules* or *boolean modules over idempotent semirings*. It is also possible to find a one-sorted signature closely related to that of Core XPath: a brief discussion is provided in Section 3.2 below.

2.1.4. The issue of unique node labels

Recall that in "real" XML trees, unlike the ones that we are using, each node satisfies exactly one label. In order to obtain completeness for this more restricted semantics, it would suffice to add a further axiom scheme:

$$v \wedge v' \equiv \bot \quad \text{for distinct } v, v' \in \mathsf{lab}$$

In what follows, however, we find it more convenient not to add the above to our list of basic axioms. Thus, one may think of labels in the present setting as modelling both XML tag names and attribute-value pairs.

2.2. Axis-specific axioms

All axioms presented up to now are valid on arbitrary structures, not only on finite trees. Table 4 introduces specific axioms for all single axis fragments. Recall that in order to axiomatize Core XPath(a) for arbitrary a, we need only a-instances of corresponding axioms.

TransAx1, also known as the *Löb axiom*, is valid on transitive structures which are *well-founded*, i.e., contain no infinite ascending R_{s^+}-chains and no R_{s^+}-cycles. It is a powerful axiom. Modulo basic axioms given in Table 3, all the node expres-

Table 6
Examples of equivalences derivable from basic axioms.

Der1	$\phi \vee \psi$	$\equiv$	$\psi \vee \phi$
Der2	$\phi \vee (\psi \vee \chi)$	$\equiv$	$(\phi \vee \psi) \vee \chi$
Der3	$A[\phi]$	$\equiv$	$A/.[\phi]$
Der4	$A[\mathsf{true}]$	$\equiv$	A
Der5	$A[\mathsf{false}]$	$\equiv$	$\bot$
Der6	$\langle A[\phi \vee \psi]\rangle$	$\equiv$	$\langle A[\phi]\rangle \vee \langle A[\psi]\rangle$
Der7	$(A \cup B)[\phi]$	$\equiv$	$A[\phi] \cup B[\phi]$
Der8	ϕ	$\leqslant$	true
Der9	$\phi \vee \neg\phi$	$\equiv$	true
Der10	A	$\equiv$	$A[\phi] \cup A[\neg\phi]$
Der11	$.[\langle A\rangle]/A$	$\equiv$	A
Der12	$\langle A/B\rangle$	$\leqslant$	$\langle A\rangle$
Der13	$\langle A[\mathsf{false}]\rangle$	$\equiv$	false
Der14	$\langle A[\phi \wedge \psi]\rangle \wedge \neg\langle A[\psi]\rangle$	$\equiv$	false
Der15	$\langle A[\phi]\rangle \wedge \neg\langle A[\psi]\rangle$	$\leqslant$	$\langle A[\phi \wedge \neg\psi]\rangle$
Der16	$A[\phi]/.[\neg\phi]$	$\equiv$	$\bot$
Der17	$A/\bot$	$\equiv$	$\bot$
	$\bot/A$	$\equiv$	$\bot$
Der18	$A[\phi][\neg\phi]$	$\equiv$	$\bot$
Der19	$A[\phi \wedge \psi][\neg\phi]$	$\equiv$	$\bot$
Der20	$A[\phi][\psi][\neg\phi \vee \neg\psi]$	$\equiv$	$\bot$
Der21	$A[\phi][\psi]$	$\equiv$	$A[\phi \wedge \psi]$
Der22	$A[\phi \wedge \psi]$	$\equiv$	$A[\phi]/.[\psi]$
Der23	$\phi \wedge \psi$	$\equiv$	$\langle .[\phi]/.[\psi]\rangle$
Der24	$.[\langle A\rangle]/.[\phi]/A$	$\equiv$	$.[\phi]/A$

sion equivalences derivable from TransAx2 are already derivable from TransAx1 (see [3,33] for more information). TransAx2, by itself, forces the transitivity of R_{s^+}. LinNTAx and TransAx3 are axioms of *functionality* and *linearity*, natural both from the point of view of Relation/Dynamic Algebras and modal logic (cf. logics **Alt.1** and **GL.3**).

2.3. Axioms for full Core XPath

We will now propose an axiomatization for full Core XPath. Table 5 presents a new group of axioms governing interactions between axes; the symbol s^{-1} denotes the converse of s. The new group of axioms we add are *tree axioms* given in Table 5. They are analogous to those for the *logic of finite trees* (*LOFT*) in [4] but are, we believe, more transparent. An interested reader can refer to Table 8 below, which shows the original axioms of [4] rewritten as simple node expressions.

TransAx1 is already familiar from single axis fragments. TreeAx1 is a Kleene algebra axiom [16,17] corresponding to an axiom of *propositional dynamic logic* (*PDL*) (cf. *Mix* in [7]). TreeAx2, TreeAx3 and TreeAx4 force that $R_\rightarrow$ is the converse of $R_\leftarrow$, $R_\uparrow$ is the converse of $R_\downarrow$ and that $R_\uparrow$, $R_\leftarrow$ and $R_\rightarrow$ are partial functions. TreeAx3 together with TreeAx4 in addition ensure proper interplay between horizontal and vertical axes.

2.4. Rules and derivations

Definition 1. For P, Q both path expressions or both node expressions, we say that $P \equiv Q$ is *derivable* from a given set of axioms if it can be obtained from them using the standard rules of equational logic:

- $P \equiv P$.
- If $P \equiv Q$ then $Q \equiv P$.
- If $P \equiv Q$ and $Q \equiv R$, then $P \equiv R$.
- If $P \equiv Q$ and R' is obtained from R by replacing some occurrences of P by Q, then $R \equiv R'$.

We will say that an expression Γ is *consistent* relative to a given set of axioms if $\Gamma \equiv \bot$ is not derivable. An expression Γ is *provably equivalent* to Δ relative to a given set of axioms if $\Gamma \equiv \Delta$ is derivable from these axioms.

The above is the standard definition of derivability used in universal algebra. However, there is an equivalent presentation which shows better the connection with query rewriting. Given two Core XPath node (path) expressions Γ, Δ, a *derivation* from Γ to Δ is a sequence of node (path) expressions $\Gamma_1, \ldots, \Gamma_n$ such that $\Gamma_1 = \Gamma$, $\Gamma_n = \Delta$, and for each $i < n$, Γ_{i+1} is obtained from Γ_i by replacing an occurrence of a subexpression Θ in Γ_i by Θ', provided that $\Theta \equiv \Theta'$ or $\Theta' \equiv \Theta$ is an instance of one of the axioms. It is not hard to see that $\Gamma \equiv \Delta$ is a derivable equivalence scheme if and only if there exists a derivation from Γ to Δ (and also vice versa, of course). In the "if" direction, the standard rules of equational logic,

Table 7
Translation of Core XPath(A) node expressions into simple node expressions.

v^{s}	$:=$	v	$.^{\mathsf{s}}(\phi)$	$:=$	ϕ
$(\neg\phi)^{\mathsf{s}}$	$:=$	$\neg\phi^{\mathsf{s}}$	$\mathsf{a}^{\mathsf{s}}(\phi)$	$:=$	$\langle \mathsf{a}[\phi]\rangle$
$(\phi \vee \psi)^{\mathsf{s}}$	$:=$	$\phi^{\mathsf{s}} \vee \psi^{\mathsf{s}}$	$(A \cup B)^{\mathsf{s}}(\phi)$	$:=$	$A^{\mathsf{s}}(\phi) \vee B^{\mathsf{s}}(\phi)$
$\langle A\rangle^{\mathsf{s}}$	$:=$	$A^{\mathsf{s}}(\mathsf{true})$	$(A[\psi])^{\mathsf{s}}(\phi)$	$:=$	$A^{\mathsf{s}}(\psi^{\mathsf{s}} \wedge \phi)$
			$(A/B)^{\mathsf{s}}(\phi)$	$:=$	$A^{\mathsf{s}}(B^{\mathsf{s}}(\phi))$

intuitively, can be read as saying that such derivation sequences are sound in the sense that they preserve equivalence. In the "only if" direction, it's easy to see that derivability by means of derivation sequences satisfies transitivity, symmetry, reflexivity and can be applied inside the context of another expression. As some of our axioms are formulated using $\leqslant$, let us recall that this is just an abbreviation for an equivalence of a specific form. Hence, the fact that $\phi \leqslant \psi$ is an axiom means that $\phi \vee \psi$ can be replaced by ψ in any derivation sequence.

Lemma 2. *All equivalences in Table 6 can be derived from those in Table 3.*

Proof. See Appendix A. □

In the remainder of this section we prove our first important result concerning syntactic derivability: a normal form theorem for node expressions. The notions of derivability, consistency and provable equivalence are considered relative to axioms in Table 3. It follows that this result does not rely on any axiom valid only for particular axes or only on finite trees; it would remain valid under arbitrary interpretation of axis constants. When derivability and related notions are mentioned in the context of Core XPath(a) for any a, the only axis constant allowed in instances of axiom schemes used in the derivation is a itself, i.e., in derivations for Core XPath(a), one is allowed to use only a-instances of axioms. In the same way, we will talk about derivable equivalence schemes. Derivations of equivalence schemes from equivalence schemes should be seen as patterns of infinitely many derivations of corresponding axiom instances.

2.5. Simple node expressions

This subsection defines a proper subclass of node expressions which is sufficiently rich to represent all node expressions. First, a few words of introduction are necessary.

Throughout the paper, we use implicitly the intimate relationship between node expressions of Core XPath and formulas of modal logic. In fact, if we take *modal logic* in a sufficiently broad sense, not only node expressions, but the whole Core XPath can be treated as a specific modal language: see [20,28,30] for a detailed discussion. In this paper, however, we use the term *modal logic* in a conservative sense—to denote systems which enrich propositional boolean logic with a supply of unary *operators* (operations preserving the bottom element and distributing over unary joins) and whose variables are interpreted as sets of nodes. Rather than providing an explicit definition, we will single out a subclass of those node expressions which correspond directly to modal formulas over a given supply of operators A. They will be called *simple node expressions* of Core XPath(A):

$$\mathsf{siNode(A)} := \mathsf{true} \mid v \mid \langle \mathsf{a}[\,\mathsf{siNode}]\rangle \mid \neg\mathsf{siNode} \mid \mathsf{siNode} \vee \mathsf{siNode}$$

where $v \in \mathsf{lab}$ and $\mathsf{a} \in \mathsf{A}$. Readers familiar with modal logic will realize that our $\langle \mathsf{a}[v]\rangle$ is just the modal formula $\Diamond_{\mathsf{a}} v$, where $\Diamond_{\mathsf{a}}$ is interpreted by R_{a}. For more on the connection with modal logic, see [28] and further comments in the present paper.

Simple node expressions are as expressive as the whole set of node expressions; the remaining node expressions (but not *path* expressions!) can be thus considered syntactic sugar. To prove this, we provide a translation $(\cdot)^{\mathsf{s}} : \mathsf{NodeEx(A)} \mapsto \mathsf{siNode(A)}$ which is constant for elements of siNode(A). This mapping uses an auxiliary mapping $(\cdot)^{\mathsf{s}} : \mathsf{PathEx(A)} \mapsto (\mathsf{siNode(A)} \mapsto \mathsf{siNode(A)})$ assigning to every path expression a unary function defined on simple node expressions. As domains of both mappings are disjoint, we use the same symbol with no risk of confusion. Their definitions are given in Table 7.

Lemma 3. *Every* Core XPath *node expression is provably equivalent to a simple node expression—namely, to its $(\cdot)^{\mathsf{s}}$-translation.*

Recall that by "provably equivalent" we mean derivability from the axioms in Table 3.

Proof. For the purpose of induction, we prove in fact a stronger statement:

For every $A \in \mathsf{PathEx}$ and for every $\phi \in \mathsf{NodeEx}$, $\langle A[\phi]\rangle \equiv A^{\mathsf{s}}(\phi)$ and $\phi \equiv \phi^{\mathsf{s}}$ are provable.

Inductive steps for node expressions are obvious, hence we focus only on inductive steps for path expressions.

- $A = .$: by NdAx2.
- $A = \mathsf{a} \in \mathsf{siAxis}$: by definition of $(\cdot)^{\mathsf{s}}$.
- $A = B \cup C$:

$$\begin{aligned} \langle (B \cup C)[\phi] \rangle &\equiv \langle B[\phi] \cup C[\phi] \rangle && \text{by Der7} \\ &\equiv \langle B[\phi] \rangle \vee \langle C[\phi] \rangle && \text{by NdAx3} \\ &\equiv B^{\mathsf{s}}(\phi) \vee C^{\mathsf{s}}(\phi) && \text{by IH} \\ &\equiv (B \cup C)^{\mathsf{s}}(\phi) && \text{by definition of } (\cdot)^{\mathsf{s}} \end{aligned}$$

- $A = B/C$:

$$\begin{aligned} \langle (B/C)[\phi] \rangle &\equiv \langle B/C[\phi] \rangle && \text{by PrAx4} \\ &\equiv \langle B[\langle C[\phi] \rangle] \rangle && \text{by NdAx4} \\ &\equiv B^{\mathsf{s}}(C^{\mathsf{s}}(\phi)) && \text{by IH} \\ &\equiv (B/C)^{\mathsf{s}}(\phi) && \text{by definition of } (\cdot)^{\mathsf{s}} \end{aligned}$$

- $A = B[\psi]$:

$$\begin{aligned} \langle B[\psi][\phi] \rangle &\equiv \langle B[\psi^{\mathsf{s}}][\phi] \rangle && \text{by IH on NodeEx} \\ &\equiv \langle B[\psi^{\mathsf{s}} \wedge \phi] \rangle && \text{by Der21} \\ &\equiv B^{\mathsf{s}}(\psi^{\mathsf{s}} \wedge \phi) && \text{by IH on PathEx} \\ &\equiv (B[\psi])^{\mathsf{s}}(\phi) && \text{by definition of } (\cdot)^{\mathsf{s}} \qquad \square \end{aligned}$$

We hope that this proof gives the reader good insight into how Birkhoff-style derivations look like. In what follows, we are going either to delegate such detailed proofs to Appendix A or just hint at main steps in the derivation, leaving the details to the reader.

2.6. Normal forms for simple node expressions

In this subsection, we are concerned with simple node expressions for single axis fragments of the form Core XPath(a) for a given a. We are going to define a normal form for such expressions motivated by analogous results in modal logic (see [6,15,23]). These normal forms are going to play a crucial rôle in completeness proofs below. They can be thought of as total descriptions of nodes accessible from the context node in a fixed number of steps along the axis a.

Define the *degree of a simple node expression* as the maximal number of nested occurrences of $\langle \mathsf{a}[\cdot] \rangle$; i.e., the degree of a label is 0, the degree of a Boolean expression is the maximum of degrees of its Boolean components and the degree of $\langle \mathsf{a}[\phi] \rangle$ is the degree of ϕ plus one. The set of all simple node expressions of degree at most n is denoted by $\mathsf{SNE}^{\mathsf{a}}_n$.

From now on, unless stated otherwise, we keep the number of labels fixed as m, that is, we assume all expressions use only the labels among $v_1, \ldots, v_m$. For any finite set $s \subseteq \{1, \ldots, k\}$, define $k \setminus s$ to be $\{1, \ldots, k\} - s$. Let

$$\mathsf{NFN}^{\mathsf{a}}_0 = \left\{ \bigwedge_{i \in s} v_i \wedge \bigwedge_{i \in m \setminus s} \neg v_i \;\middle|\; s \subseteq \{1, \ldots, m\} \right\}.$$

Assume $\mathsf{NFN}^{\mathsf{a}}_i$ are defined for all i smaller than $n \geqslant 1$. Let $f(n)$ be the cardinality of $\bigcup_{i<n} \mathsf{NFN}^{\mathsf{a}}_i$. Fix an enumeration of this set $\alpha_1, \ldots, \alpha_{f(n)}$ and define the set $\mathsf{NFN}^{\mathsf{a}}_n$ of all expressions of the form

$$\phi^{\beta,s} := \beta \wedge \bigwedge_{i \in s} \langle \mathsf{a}[\alpha_i] \rangle \wedge \bigwedge_{i \in f(n) \setminus s} \neg \langle \mathsf{a}[\alpha_i] \rangle$$

for arbitrary $\beta \in \mathsf{NFN}^{\mathsf{a}}_0$ and $s \subseteq \{1, \ldots, f(n)\}$ s.t. $\phi^{\beta,s}$ is consistent relative to a-instances of all axioms for a given single axis fragment. β can be thought of as the configuration of labels valid at the present node, while the remaining conjuncts describe the situation in nodes reachable in no more than n steps along the axis a. For any ϕ of the above form and for any $\alpha_i \in \bigcup_{j<n} \mathsf{NFN}^{\mathsf{a}}_j$, we say that α_i *is positive* in ϕ if $i \in s$, and otherwise α_i *is negative* in ϕ.

Lemma 4.

- *Every simple node expression in* $\mathsf{SNE}^{\mathsf{a}}_n$ *is provably equivalent to a disjunction of (zero or more) elements of* $\mathsf{NFN}^{\mathsf{a}}_n$.
- *For every pair of distinct elements* $\phi, \psi \in \mathsf{NFN}^{\mathsf{a}}_n$, $\phi \wedge \psi$ *is inconsistent.*

Proof *(Sketch).* The proof is a generalization of the proof of the normal form theorem for classical propositional logic. Recall that every formula of classical calculus can be represented as disjunction of conjunctions of literals: atoms and their negations. In this case, the collection of literals has to be extended with normal forms of strictly lower degree. The equivalence scheme Der6 in Table 6 is crucially used in distributing the disjunctions locked inside test expressions. See [6, Theorem 1] or [15, Theorem 2.7] for different notational variants of the same proof (the result proved in the latter paper is in fact even more general than the one proved here). □

Corollary 5. *For every* $\phi \in \mathsf{NFN}^{\mathsf{a}}_{n+1}$, *there is exactly one* $\phi^{\mathsf{d}} \in \mathsf{NFN}^{\mathsf{a}}_n$ *s.t.* ϕ *and* ϕ^{d} *are consistent.*

Proof. $\mathsf{NFN}^{\mathsf{a}}_n$ is a subset of $\mathsf{SNE}^{\mathsf{a}}_{n+1}$, thus by Lemma 4 each element of $\mathsf{NFN}^{\mathsf{a}}_n$ is equivalent to a disjunction of elements of $\mathsf{NFN}^{\mathsf{a}}_{n+1}$. However, by Lemma 4, elements of $\mathsf{NFN}^{\mathsf{a}}_n$ are mutually inconsistent, hence any element of $\mathsf{NFN}^{\mathsf{a}}_{n+1}$ can appear as a disjunct for only one element of $\mathsf{NFN}^{\mathsf{a}}_n$. □

In [6], ϕ^{d} is called the *correlate* of ϕ and in [23]—the *derivative* of ϕ. We adopt the latter term. Intuitively, if ϕ can be thought of as a total description of nodes accessible from the context node in no more than $n+1$ steps along the axis a, ϕ^{d} restricts the information provided by ϕ to obtain a complete description of nodes accessible from the context node in no more than n steps along the same axis. In fact, ϕ^{d} can be obtained from ϕ by a straightforward syntactic operation: removing the deepest nesting level in every conjunct. For example, in a language with one variable

$$(v_1 \wedge \langle{\downarrow^+}[v_1]\rangle \wedge \langle{\downarrow^+}[\neg v_1]\rangle)^{\mathsf{d}} = v_1.$$

Nevertheless, the existence and uniqueness of derivative is a direct consequence of Lemma 4 (together with the restriction of the set of normal forms to consistent ones) and its specific syntactic shape is irrelevant for our purposes.

3. Completeness results for single axis fragments

We are ready to present completeness proofs for specific axes. In each of these proofs, the notions of *provability* and *consistency* are relative to the set of axioms in the formulation of the corresponding completeness theorem; we will not repeat this disclaimer. We begin with the descendant axis. In proofs for remaining axes later on, we can focus on differences (and/or similarities) with the proof for Core XPath($\downarrow^+$).

3.1. Completeness for Core XPath($\downarrow^+$)

Theorem 6 *(Node completeness for* Core XPath($\downarrow^+$)*). A* Core XPath($\downarrow^+$) node *equivalence is valid iff it is derivable from* $\downarrow^+$*-instances of the axioms in Table 3 and* TransAx1, TransAx2 *in Table 4.*

Theorem 7 *(Path completeness for* Core XPath($\downarrow^+$)*). A* Core XPath($\downarrow^+$) path *equivalence is valid iff it is derivable from* $\downarrow^+$*-instances of the axioms in Table 3 and* TransAx1, TransAx2 *in Table 4.*

While Theorem 6 can be derived from known results in modal literature, Theorem 7 seems new. As the proof of the latter we are going to give builds on a specific technique used in the proof of the former, we are going to give uniform proofs for both results. Our proof of Theorem 6 is analogous to [23, Theorem 6.2].

For the sake of both proofs, we represent XML trees as tuples $T = (N, R_{\downarrow^+}, L)$, where $R_{\downarrow^+}$ is the descendant axis. This convention allows to phrase some lemmas more succinctly. Note that, since Core XPath($\downarrow^+$) does not provide means to refer to the sibling order, the latter may be chosen arbitrarily.

For $\phi, \psi \in \mathsf{NFN}^{\downarrow^+}_n$, we write $\phi \preceq_{\downarrow^+} \psi$ if the following two conditions hold:

- every element of $\bigcup_{i<n} \mathsf{NFN}^{\downarrow^+}_i$ positive in ψ is positive in ϕ and
- the derivative of ψ is also positive in ϕ.

Intuitively, $\phi \preceq_{\downarrow^+} \psi$ means that a node where ϕ holds can be an ancestor of a node where ψ holds. Recall that α being positive in ϕ means that the validity of ϕ implies the validity of $\langle{\downarrow^+}[\alpha]\rangle$, i.e., that every node satisfying ϕ has a $\downarrow^+$-successor where α is valid. Conversely, α being negative in ϕ means that no node satisfying ϕ has a $\downarrow^+$-successor where α is valid.

Note that $\preceq_{\downarrow^+}$ is in general not necessarily *an order*: there exist non-trivial $\preceq_{\downarrow^+}$-cycles. However, we have the following

Lemma 8. *$\preceq_{\downarrow^+}$ is a transitive relation.*

Proof. Immediate from the definition $\preceq_{\downarrow^+}$. Observe that those normal forms which are negative in a given ϕ must stay negative in all $\preceq_{\downarrow^+}$-successors of ϕ. Hence, if ψ^{d} is positive in a $\preceq_{\downarrow^+}$-successor of ϕ, it is positive in ϕ itself. $\square$

Here are some examples illustrating this notion of precedence:

- $\phi \preceq_{\downarrow^+} \psi$ for any $\phi, \psi \in \mathsf{NFN}_0^{\downarrow^+}$,
- in a language with just one variable:

$$v_1 \wedge \langle\downarrow^+[v_1]\rangle \wedge \langle\downarrow^+[\neg v_1]\rangle \preceq_{\downarrow^+} \neg v_1 \wedge \neg\langle\downarrow^+[v_1]\rangle \wedge \langle\downarrow^+[\neg v_1]\rangle,$$

  but the converse does not hold.

If for $\phi, \psi \in \mathsf{NFN}_n^{\downarrow^+}$, we have $\phi \preceq_{\downarrow^+} \psi$ and some $\chi \in \bigcup_{i<n} \mathsf{NFN}_i^{\downarrow^+}$ is positive in ϕ but negative in ψ, then we write $\phi \prec_{\downarrow^+} \psi$. It follows from the definition that $\prec_{\downarrow^+}$ is not only transitive, but also *well-founded*: it contains no infinite descending chains and no loops.

Lemma 9. *Every element of $\mathsf{NFN}_n^{\downarrow^+}$ is satisfiable.*

Proof *(Sketch).* Take any $\phi \in \mathsf{NFN}_n^{\downarrow^+}$. We construct an XML tree satisfying ϕ at the root as follows. The domain N of our XML tree consists of all sequences of elements of $\mathsf{NFN}_n^{\downarrow^+}$ of the form $(\beta_1, \ldots, \beta_k)$, with $\beta_1 = \phi$ and for any $j < k$, $\beta_j \prec_{\downarrow^+} \beta_{j+1}$. Note that there are only finitely many such sequences and that for any $i < j$, $\beta_i \prec_{\downarrow^+} \beta_{j+1}$ as well. For $x, y \in N$, define $xR_{\downarrow^+}y$ to hold if the sequence x is an initial segment of the sequence y (recall the convention about trees in the present proof). Finally, the labelling function L labels the node $(\beta_1, \ldots, \beta_k)$ with v if v is positive in β_k. It can be shown by induction that the root of the XML tree obtained this way, i.e., (ϕ), indeed satisfies ϕ. The key observation that all normal forms positive in ϕ are witnessed at some $\prec_{\downarrow^+}$-successor follows from axiom TransAx1. We call this tree the *canonical tree* of ϕ. The reader is encouraged to compare this proof with [6, Theorems 4 and 5] or [23, Theorem 6.2]. $\square$

Theorem 6 now follows:

Proof of Theorem 6. We restrict ourselves to the difficult direction. Suppose that $\phi \equiv \psi$ is valid, where ϕ, ψ are arbitrary Core XPath($\downarrow^+$) node expressions. By Lemma 3 and the first item of Lemma 4, for large enough n, ϕ is provably equivalent to some disjunction ϕ' of $\mathsf{NFN}_n^{\downarrow^+}$ expressions, and ψ is equivalent to some disjunction ψ' of $\mathsf{NFN}_n^{\downarrow^+}$ expressions. It follows by the remaining item of Lemma 4 and by Lemma 9 that ϕ' and ψ' must be identical (up to the ordering of the disjuncts): if one contains a disjunct which does not appear in the other, this disjunct is satisfiable and wherever it is satisfied, no other disjunct may hold, a contradiction. Hence, ϕ' and ψ' are provably equivalent, and therefore so are ϕ and ψ. $\square$

Next, we will proceed to prove Theorem 7.

Definition 10. $\mathsf{NFP}_n^{\downarrow^+}$ is the set of consistent path expressions of the form

$$S = .[\beta_1]/\downarrow^+[\beta_2]/\cdots/\downarrow^+[\beta_\ell],$$

where $\ell \geqslant 1$, each $\beta_i \in \mathsf{NFN}_n^{\downarrow^+}$, and $\beta_i \preceq_{\downarrow^+} \beta_j$ for $i < j$.

Note that we use the weak relation $\preceq_{\downarrow^+}$ here, not the strict order $\prec_{\downarrow^+}$ used in the construction of the models in the node completeness proof.

Lemma 11. *For every path expression A, there exists suitably large n s.t. for every $n' \geqslant n$, A is provably equivalent to a sum of elements of $\mathsf{NFP}_{n'}^{\downarrow^+}$.*

Proof *(Sketch).* Repeated use of Lemma 4, TransAx2 and some auxiliary equivalences in Tables 3 and 6. In some more detail: remove all non-leading occurrences of "." (i.e., occurrences preceded by "/") outside the scope of any node subexpression using Der3 and Der22. Fix n to be the sum of the highest degree of a subexpression occurring in A and the number of occurrences of "/" operator in A. Then all maximal node subexpressions of A (i.e., those which themselves are not proper subexpressions of any other node subexpression of A) can be rewritten as disjuncts of elements of $\mathsf{NFP}_n^{\downarrow^+}$ by Lemma 4. Distribute all sums and disjunctions wherever possible. The inconsistency of these summands where $\beta_i \preceq_{\downarrow^+} \beta_{i+1}$ does not hold for some i is now a direct consequence of Der11 and Der22, whereas for $j > i + 1$ one uses in addition TransAx2. $\square$

We are ready to introduce the construction of canonical models associated with normal forms of path expression using an auxiliary merging construction for XML trees.

Definition 12. Given two XML trees $T_1 = (N, R_{\downarrow^+}, L)$ and $T_2 = (N', R'_{\downarrow^+}, L')$ with roots r and r', respectively, and N, N' disjoint, we define their *transitive root union* $T_1 \rhd T_2$ as the XML tree $(N \cup N', R_{\downarrow^+} \cup R'_{\downarrow^+} \cup (\{r\} \times N'), L \cup L')$. That is, the root of the second becomes a child of the root of the first.

For any $S \in \mathsf{NFP}_n^{\downarrow^+}$ of the form

$$S = .[\beta_1]/\downarrow^+[\beta_2]/\cdots/\downarrow^+[\beta_\ell],$$

we define the *canonical tree* of S as the structure

$$T = T_1 \rhd \bigl(T_2 \rhd (\cdots \rhd T_\ell)\cdots\bigr),$$

where each T_i is the canonical tree of β_i as defined in the proof of Lemma 9 above.

Lemma 13. *Let $S \in \mathsf{NFP}_n^{\downarrow^+}$ and its canonical tree T be as in Definition 12 above and for each $i \leqslant \ell$, let r_i be the root of the tree T_i. Then $(r_1, r_\ell) \in [\![S]\!]_{\mathsf{PExpr}}$. Moreover, for any $S' = .[\beta'_1]/\downarrow^+[\beta'_2]/\cdots/\downarrow^+[\beta'_{\ell'}] \in \mathsf{NFP}_n^{\downarrow^+}$, $(r_1, r_\ell) \in [\![S']\!]_{\mathsf{PExpr}}$ iff $(\beta'_1, \ldots, \beta'_{\ell'})$ is a subsequence of $(\beta_1, \ldots, \beta_\ell)$ s.t. $\beta_1 = \beta'_1$ and $\beta'_{\ell'} = \beta_\ell$.*

Proof. It is enough to prove the "moreover" part. The "if" direction is by direct verification (recall that if $\beta_i \preceq_{\downarrow^+} \beta_j$, then every formula negative in β_i is also negative in β_j). For the converse, assume the contrary and recall that by Lemma 9, no two distinct elements of $\mathsf{NFN}_n^{\downarrow^+}$ can be true at the same point. Thus, if β'_1 is distinct from β_1 or $\beta'_{\ell'}$ from β_ℓ, there is nothing to prove. Let $i \geqslant 2$ be the smallest number s.t. $(\beta'_1 = \beta_1, \beta'_2, \ldots, \beta'_i)$ is not a subsequence of $(\beta_1, \beta_2, \ldots, \beta_\ell)$. That is, $(\beta'_1 = \beta_1, \beta'_2, \ldots, \beta'_{i-1}) = (\beta_1, \beta_{g(2)}, \ldots, \beta_{g(i-1)})$ for some strictly increasing g, but $\beta'_i = \beta_j$ for no j s.t. $g(i-1) < j \leqslant \ell$. For any $1 < j < i$, we choose $g(j)$ to be *minimal* so that for no j' properly contained between $g(j-1)$ and $g(j)$, $\beta_{j'}$ is equal to β'_j. So it means that $\beta'_1 = \beta_1, \beta'_2 = \beta_{g(2)}, \ldots, \beta'_{i-1} = \beta_{g(i-1)}$ are true at, respectively, $r_1, r_{g(2)}, \ldots, r_{g(i-1)}$ (and no subsequence of $(r_1, r_2, \ldots, r_{g(i-1)-1})$ makes them valid in this order) but for no $j > g(i-1)$, β'_i does hold at r_j. It follows that $(r_1, r_\ell) \notin [\![S']\!]_{\mathsf{PExpr}}$. $\square$

Lemma 14. *For any*

$$\left.\begin{array}{l} S' = .[\beta'_1]/\downarrow^+[\beta'_2]/\cdots/\downarrow^+[\beta'_{\ell'}] \\ S_1 = .[\beta^1_1]/\downarrow^+[\beta^1_2]/\cdots/\downarrow^+[\beta^1_{\ell(1)}] \\ \quad\cdots \\ S_k = .[\beta^k_1]/\downarrow^+[\beta^k_2]/\cdots/\downarrow^+[\beta^k_{\ell(k)}] \end{array}\right\} \in \mathsf{NFP}_n^{\downarrow^+},$$

S' is contained in $S_1 \cup \cdots \cup S_k$ iff for some $i \leqslant k$,

$$(*) \qquad (\beta^i_1, \ldots, \beta^i_{\ell(i)}) \text{ is a subsequence of } (\beta'_1, \ldots, \beta'_{\ell'}) \quad \text{s.t.} \quad \beta'_1 = \beta^i_1 \text{ and } \beta'_{\ell'} = \beta^i_{\ell(i)}.$$

Proof. The "if" direction follows by a direct calculation. Conversely, Lemma 13 implies that if the condition $(*)$ above holds for no $i \leqslant k$, then $(r_1, r_{\ell'})$ in the canonical tree belongs to $[\![S']\!]_{\mathsf{PExpr}}$, but not to $[\![S_1]\!]_{\mathsf{PExpr}} \cup \cdots \cup [\![S_k]\!]_{\mathsf{PExpr}}$. Thus, we have a countermodel for containment. $\square$

Finally, we prove Theorem 7:

Proof of Theorem 7. Follows from Lemmas 11 and 14; the reasoning is in fact analogous to the proof of Theorem 6 above. Given any equivalence, we can rewrite both sides as sums of elements of $\mathsf{NFP}_n^{\downarrow^+}$ for some suitably high n, then use Lemma 14 to show that if one side is not provably contained in the other, there exists a countermodel. $\square$

3.2. *Completeness for* Core XPath(↓)

Similar results can be proved for Core XPath(↓):

Theorem 15 *(Node completeness for* Core XPath(↓)*). A* Core XPath(↓) node *equivalence is valid iff it is derivable from ↓-instances of the axioms in Table 3.*

Theorem 16 *(Path completeness for* Core XPath(↓)*). A* Core XPath(↓) path *equivalence is valid iff it is derivable from ↓-instances of the axioms in Table 3.*

In other words, there are no Core XPath($\downarrow$)-specific validities.

Proof of Theorems 15 and 16. (*Sketch*) We only highlight the most important differences with the proofs of Theorems 6 and 7. We define a (non-transitive) relation $\prec_\downarrow$ not just on $\mathsf{NFN}^{\downarrow}_n$, but on the whole set $\bigcup_{i\leqslant n}\mathsf{NFN}^{\downarrow}_i$, relating normal form node expressions of degree i to ones of degree $i-1$. More precisely, we say that $\phi \prec_\downarrow \phi'$ if for some $i \leqslant n$, $\phi \in \mathsf{NFN}^{\downarrow}_i$, $\phi' \in \mathsf{NFN}^{\downarrow}_{i-1}$ and ϕ' is positive in ϕ. Canonical trees based on this order are defined similarly as in the proof of Lemma 9, this time using the child relation rather than the descendant relation. That is, we make $(\beta_1,\dots,\beta_k)$ a parent of $(\beta_1,\dots,\beta_k,\beta_{k+1})$ in the canonical tree. In the normal form for path expressions, the degree of node expressions in the sequence decreases. The definition of canonical trees for normal form path expressions does not require essential changes and the proof of an analogue of Lemma 13 is in fact simplified by the lack of transitivity. $\square$

The proof method used for Theorem 15 corresponds to the one used for [6, Theorem 3] in modal logic. Theorem 16 is closely related to a completeness result for *dynamic relation algebras with union* in [12, Theorem 4.9 and Section 5]. See [20,30] for more details on the connection between Core XPath and this algebraic language.

3.3. Completeness for transitive linear axes

Theorem 17 *(Node completeness for transitive linear axes). A* Core XPath(s^+) ($\mathsf{s} \in \{\rightarrow^+, \leftarrow^+, \uparrow^+\}$) node *equivalence is valid iff it is derivable from* s^+*-instances of axioms in Table 3 and TransAx1, TransAx2 and TransAx3 in Table 4.*

Theorem 18 *(Path completeness for transitive linear axes). A* Core XPath(s^+) ($\mathsf{s} \in \{\rightarrow^+, \leftarrow^+, \uparrow^+\}$) path *equivalence is valid iff it is derivable from* s^+*-instances of axioms in Table 3 and TransAx1, TransAx2 and TransAx3 in Table 4.*

For $\phi,\phi' \in \mathsf{NFN}^{\mathsf{s}^+}_n$, we define $\phi \preceq_{\mathsf{s}^+} \phi'$ in the same way as for $\downarrow^+$. We can now define the set $\mathsf{NFP}^{\mathsf{s}^+}_n$ analogously to Definition 10—recall that this definition included only *consistent* path normal forms—and an analogue of Lemma 11 still holds. However, these normal forms for path expressions are not "saturated" enough to construct canonical linearly ordered models out of them. We therefore define a proper subset $\mathsf{satNFP}^{\mathsf{s}^+}_n \subseteq \mathsf{NFP}^{\mathsf{s}^+}_n$.

Definition 19. $\mathsf{satNFP}^{\mathsf{s}^+}_n$ is the set of elements of $\mathsf{NFP}^{\downarrow^+}_n$ of the form

$$S = .[\beta_1]/\mathsf{s}^+[\beta_2]/\cdots/\mathsf{s}^+[\beta_\ell],$$

where $\ell \geqslant 1$, each $\beta_i \in \mathsf{NFN}^{\mathsf{s}^+}_n$, $\beta_i \preceq_{\mathsf{s}^+} \beta_j$ for $i<j$ and for every $i<\ell$ and every α positive in β_i it is either the case that α is positive in β_ℓ or for some $j>i$,

- $\alpha \wedge \beta_j$ is consistent and
- α is negative in β_j.

As β_j is a normal form of degree higher than α, the consistency of $\alpha \wedge \beta_j$ is equivalent to α being the kth derivative of β_j for a suitable k. Intuitively, β_j is the last node where α holds and everywhere further down the axis s^+, α is refuted. The fact that it is always possible to find such a s^+-maximal node with respect to any α which holds somewhere along a s^+-path is exactly the meaning of TransAx1 axiom for s^+. TransAx3, on the other hand, ensures s^+-paths are non-branching: if $\langle \mathsf{s}^+[\alpha]\rangle$ holds at some node on a s^+-path, α itself must hold somewhere further along the same path. Together, the two axioms play a crucial rôle in the proof of the following result:

Lemma 20. *For* $\mathsf{s}^+ \in \{\uparrow^+, \rightarrow^+, \leftarrow^+\}$*, every element of* $\mathsf{NFP}^{\mathsf{s}^+}_n$ *is equivalent to a sum of elements of* $\mathsf{satNFP}^{\mathsf{s}^+}_n$*. Consequently, for every path expression A of* Core XPath(s^+)*, there exists a suitably large n s.t. for every* $n' \geqslant n$*, A is equivalent to a sum of elements of* $\mathsf{satNFP}^{\mathsf{s}^+}_{n'}$.

Proof. As above, we fix an enumeration $\alpha_1,\dots,\alpha_{f(n)}$ of $\bigcup_{i<n}\mathsf{NFN}^{\mathsf{s}^+}_i$. We say S has an α_i*-defect* if there is $\beta_j \in S$ s.t. α_i is positive in β_j but negative in β_{j+1} and inconsistent with β_{j+1}. Let us choose the smallest i s.t. α_i is a defect. It follows from TransAx1 and the definition of a defect that S is equivalent to

$$.[\beta_1]/\mathsf{s}^+[\beta_2]/\cdots/\mathsf{s}^+[\beta_j \wedge \langle \mathsf{s}^+[\alpha_i \wedge \neg\langle \mathsf{s}^+[\alpha_i]\rangle]\rangle]/\mathsf{s}^+[\beta_{j+1} \wedge \neg\alpha_i \wedge \neg\langle \mathsf{s}^+[\alpha_i]\rangle]/\cdots/\mathsf{s}^+[\beta_\ell]$$

This in turn by TransAx3 is equivalent to

$$.[\beta_1]/\mathsf{s}^+[\beta_2]/\cdots/\mathsf{s}^+[\beta_j]/\mathsf{s}^+[\alpha_i \wedge \neg\langle \mathsf{s}^+[\alpha_i]\rangle]/\mathsf{s}^+[\beta_{j+1} \wedge \neg\alpha_i \wedge \neg\langle \mathsf{s}^+[\alpha_i]\rangle]/\cdots/\mathsf{s}^+[\beta_\ell]$$

and this finally is equivalent to

$$.[\beta_1]/\mathsf{s}^+[\beta_2]/\cdots/\mathsf{s}^+[\beta_j]/\mathsf{s}^+[\phi]/\mathsf{s}^+[\beta_{j+1} \wedge \neg\alpha_i \wedge \neg\langle \mathsf{s}^+[\alpha_i]\rangle]/\cdots/\mathsf{s}^+[\beta_\ell]$$

where ϕ stands for

$$\alpha_i \wedge \neg\langle \mathsf{s}^+[\alpha_i]\rangle \wedge \bigwedge_{\gamma \text{ negative in } \beta_i} \neg\langle \mathsf{s}^+[\gamma]\rangle \wedge \bigwedge_{\gamma \text{ positive in } \beta_{i+1}} \langle \mathsf{s}^+[\gamma]\rangle \wedge \langle \mathsf{s}^+[\beta_{j+1}{}^{\mathsf{d}}]\rangle.$$

The fact that we can add new conjuncts in the last step preserving equivalence follows from the validity of $.[\neg\langle \mathsf{s}^+[\gamma]\rangle]/\mathsf{s}^+[\gamma] \equiv \perp$ (an instance of PrAx1) and TransAx2. Now, Lemma 4 guarantees ϕ is equivalent to a disjunction of elements of $\mathsf{NFN}_n^{\mathsf{s}^+}$. For any $\beta_{j'}$ extending ϕ s.t.

$$S'_{j'} = .[\beta_1]/\mathsf{s}^+[\beta_2]/\cdots/\mathsf{s}^+[\beta_j]/\mathsf{s}^+[\beta_{j'}]/\mathsf{s}^+[\beta_{j+1}]/\cdots/\mathsf{s}^+[\beta_\ell]$$

is consistent (the inconsistent ones can be simply removed as summands, but if all such $S'_{j'}$ were inconsistent, the original expression would have been inconsistent as well), $S'_{j'}$ is a member of $\mathsf{NFP}_n^{\mathsf{s}^+}$ which does not have an α_i-defect and no element of $\mathsf{NFP}_n^{\mathsf{s}^+}$ extending $S'_{j'}$ without adding new points at the beginning or at the end has an α_i-defect. Thus, the α_i-defect has been fixed in all consistent $S'_{j'}$. We repeat the procedure for remaining defects. □

Proof of Theorem 17. It is clearly enough to show that every $\phi \in \mathsf{NFN}_n^{\mathsf{s}^+}$ is satisfiable. We will prove it in a slightly different manner than for Core XPath($\downarrow^+$); in fact, the proof along the present lines could have been given also for Theorem 6. Lemma 4 and Corollary 5 imply that ϕ is equivalent to $\bigvee\{\psi \in \mathsf{NFN}_{n+2}^{\mathsf{s}^+} \mid \phi = \psi^{\mathsf{dd}}\}$. Let us choose arbitrarily any ψ s.t. $\phi = \psi^{\mathsf{dd}}$. Define

$$A = \left\{\alpha \in \mathsf{NFN}_n^{\mathsf{s}^+} \;\middle|\; \alpha \text{ positive in } \psi\right\}$$

and a relation $\prec_{\mathsf{s}^+}$ on A as follows:

$$\alpha \prec_{\mathsf{s}^+} \beta \quad \text{iff} \quad \psi \wedge \langle \mathsf{s}^+[\alpha \wedge \neg\langle \mathsf{s}^+[\beta]\rangle]\rangle \text{ is inconsistent.}$$

Axioms TransAx2 and LinNTAx together with Lemma 4 guarantee that this is a linear transitive relation—total on A. In other words, any two normal forms of degree n which are positive in a chosen normal form of degree $n+2$ must be provably comparable relative to that form. TransAx1 guarantees that this relation is, in addition, antisymmetric—a well-founded order. This order, with ψ as the root, will be called a *suitable chain for* ϕ. Note that both the order and the elements occurring in it are relative to the chosen $\psi \in \mathsf{NFN}_{n+2}^{\mathsf{s}^+}$, therefore it is not uniquely determined. The reader can verify that a model whose points are the normal forms occurring in this chain and with both R_{s^+} and labelling defined in the natural way is a model for ϕ. □

Proof of Theorem 18. The canonical chain associated with S' is built by concatenating S' with a suitable chain for β_ℓ. Formulation and proofs of analogues of Lemmas 13 and 14 for elements of $\mathsf{satNFP}_n^{\mathsf{s}^+}$ and sums thereof can be now left to the reader. Observe that a reasoning analogous to the one used in the proof of Theorem 17 ensures that every $S' \in \mathsf{satNFP}_n^{\mathsf{s}^+}$ of the form

$$S' = .[\beta_1]/\mathsf{s}^+[\beta_2]/\cdots/\mathsf{s}^+[\beta_\ell]$$

contains a minimal subsequence which belongs to $\mathsf{satNFP}_n^{\mathsf{s}^+}$; this is the subsequence T of S' which contains exactly β_1, β_ℓ and those β_i's in between whose removal would create a defect. Then an element of $\mathsf{satNFP}_n^{\mathsf{s}^+}$ (provably) contains S' iff it is a subsequence of S' containing all β_i's in T. □

3.4. Completeness for non-transitive linear axes

Theorem 21 *(Node completeness for nontransitive linear axes). A* Core XPath(s) (s ∈ {→, ←, ↑}) node *equivalence is valid iff it is derivable from* s*-instances of axioms in Table 3 and LinNTAx in Table 4.*

Theorem 22 *(Path completeness for nontransitive linear axes). A* Core XPath(s) (s ∈ {→, ←, ↑}) path *equivalence is valid iff it is derivable from* s*-instances of axioms in Table 3 and LinNTAx in Table 4.*

Same remarks apply as in the case of Theorems 6 and 7: while Theorem 21 can be derived from known results in modal literature, Theorem 22 seems new. Again, we are going to give uniform proofs for both results.

Proof of Theorems 21 and 22. (*Sketch*) Just like the proof of Theorems 17 and 18 modified the proof of Theorems 6 and 7, this proof modifies the proof of Theorems 15 and 16. Again, define a (non-transitive) relation $\prec_{\mathsf{s}}$ ($\mathsf{s} \in \{\leftarrow, \rightarrow, \uparrow\}$) on the whole set $\bigcup_{i \leqslant n} \mathsf{NFN}_i^{\mathsf{s}}$ by stipulating that $\phi \prec_{\mathsf{s}} \phi'$ if for some $i \leqslant n$, $\phi \in \mathsf{NFN}_i^{\mathsf{s}}$, $\phi' \in \mathsf{NFN}_{i-1}^{\mathsf{s}}$ and ϕ' is positive in ϕ. However,

Table 8
Equivalences axiomatizing LOFT of Blackburn, Meyer-Viol, de Rijke [4].

LOFT0:	boolean axioms			
LOFT1	$\langle s[\text{false}]\rangle$	$\equiv$	false	
	$\langle s[\phi \vee \psi]\rangle$	$\equiv$	$\langle s[\phi]\rangle \vee \langle s[\psi]\rangle$	
LOFT2	ϕ	$\leqslant$	$\neg\langle s[\neg\langle s^{-1}[\phi]\rangle]\rangle$	
LOFT3	$\langle s[\neg\phi]\rangle \wedge \langle s[\phi]\rangle$	$\equiv$	false	for $s \in \{\uparrow, \leftarrow, \rightarrow\}$
LOFT4	$\langle s[\phi]\rangle \vee \langle s[\langle s^+[\phi]\rangle]\rangle$	$\equiv$	$\langle s^+[\phi]\rangle$	
LOFT5	$\neg\langle s[\phi]\rangle \wedge \langle s^+[\phi]\rangle$	$\leqslant$	$\langle s^+[\neg\phi \wedge \langle s[\phi]\rangle]\rangle$	
LOFT6	$\langle s[\text{true}]\rangle$	$\leqslant$	$\langle s^+[\neg\langle s[\text{true}]\rangle]\rangle$	
LOFT7:	TransAx1 for $\downarrow^+$ and $\rightarrow^+$			
LOFT8	$\neg\langle\downarrow[\phi]\rangle$	$\geqslant$	$\langle\downarrow[\neg\langle\leftarrow\rangle \wedge \neg\phi \wedge \neg\langle\rightarrow^+[\phi]\rangle]\rangle$	
LOFT9	$\langle\downarrow[\phi]\rangle$	$\leqslant$	$\langle\downarrow[\neg\langle\leftarrow\rangle]\rangle \wedge \langle\downarrow[\neg\langle\rightarrow\rangle]\rangle$	
LOFT10	$\neg\langle\uparrow\rangle$	$\leqslant$	$\neg\langle\leftarrow\rangle \wedge \neg\langle\rightarrow\rangle$	

using LinNTAx we can prove a very strong fact about $\prec_s$ which obviously does not hold for $\prec_\downarrow$. Namely, $\prec_s$ is a partial function, i.e., no normal form can have more than one $\prec_\downarrow$-successor. This is a consequence of the validity of $\langle s[\neg\phi]\rangle \leqslant \neg\langle s[\phi]\rangle$ and the fact that distinct normal forms of the same degree are mutually inconsistent. This also means that when we define normal forms for path expressions just like in the proof of Theorem 16, i.e., as sequences of node expressions of strictly decreasing degree, there will be no defects to be fixed, contrary to the situation we had in the proof of Theorem 18. The details are left to the reader as an exercise. □

4. Completeness for full Core XPath

We will now present a completeness result for full Core XPath based on axioms from Tables 3 and 5; the notions of derivability and consistency are now taken relative to this set of axioms. As in the previous cases, we focus first on completeness for node expressions.

Theorem 23 *(Node completeness for full* Core XPath*). A* Core XPath node *equivalence is valid iff it is derivable from the axioms given in Tables* 3 and 5.

Proof. As before, we first reduce all node expressions to simple ones using Lemma 3. Next, we observe that these simple Core XPath node expressions can be seen as notational variants of formulas of the *logic of finite trees* (*LOFT*) [4]. Finally, we show that equivalences axiomatizing LOFT are derivable when written as simple node expressions. They are given in Table 8; let us recall once again that both $\phi \leqslant \psi$ and $\psi \geqslant \phi$ abbreviate $\phi \vee \psi \equiv \psi$. See Appendix A for the derivations. Completeness is now an immediate consequence of the completeness result in [4, Theorem 42]; in fact, that theorem is exactly the completeness result for simple node expressions of Core XPath in the isomorphic modal notation. □

The node completeness result can be lifted to path expressions by introducing an extra inference rule with syntactic side conditions, which we call the *Sep* rule, and which is closely related to the *separability rule* in [24].

(**Sep**) If $\langle A[v]\rangle \equiv \langle B[v]\rangle$ and v does not occur in A and B, then $A \equiv B$.

Corollary 24 *(Non-orthodox path completeness for full* Core XPath*). A* Core XPath path *equivalence is valid iff it is derivable from the axioms in Tables* 3 and 5 *using the standard rules of equational logic plus the* Sep *rule.*

Proof. Suppose $A \equiv B$. Pick any v not occurring in A and B. Then the node expressions $\langle A[v]\rangle$ and $\langle B[v]\rangle$ are also equivalent. Hence, by Theorem 23, their equivalence is derivable (in the standard sense) from the axioms in Tables 3 and 5. A single application of the *Sep* rule now yields a derivation of $A \equiv B$. □

An in-depth discussion of disadvantages of such axiomatizations can be found in [25,34]. We should point out that the rule *Sep* as formulated here is not sound when used in combination with the axiom scheme $v \wedge v' \equiv \perp$ discussed in Section 2. The problem can be solved, but every available solution requires some additional complications, e.g., in the formulation of the *Sep* rule. This is one more reason why a purely equational axiomatization would be of interest. Note that, once the *Sep* rule is added, derivability can no longer be characterized simply by the existence of a chain of rewrite steps, as discussed in Section 2.4.

Our conjecture is that axioms presented in Tables 3, 4 and 5 (note we include single axis axioms here) provide an orthodox axiomatization for Core XPath path expressions. However, it is not clear how to provide such a completeness proof for path expressions, even given the results of [4]. Generalizing existing modal results to path equivalences without the use of non-standard rules seems a non-trivial problem, although a good choice of proof technique—such as normal modal forms in the present paper—may prove helpful.

5. Further work

We have given complete axiomatizations for all eight single axis fragments and full Core XPath. We hope that these axiomatization will be of help in obtaining sets of effective rewrite rules for query optimization in these XPath fragments.

Except for obtaining such rules and implementing them, other possible directions of research are:

- removing the non-orthodox rule *Sep* from the axiomatization of full Core XPath path equivalences;
- equationally axiomatizing *positive* fragments of Core XPath, in particular those extending the fragment studied in [2];
- equationally axiomatizing fragments of Core XPath properly contained between single axis fragments and the whole language. Of particular interest, also from a practical point of view, are fragments mixing the `following` and `descendant` axes and/or duals of both;
- investigating whether existing theorem provers and model searchers for equational logic like Waldmeister [10], Prover9 and Mace4 [21] can be adapted for automatic proving/disproving XPath equivalences (this should be rather straightforward using our results) and, more importantly, whether this would be practically useful for XPath implementers and/or programmers of query optimizers.

Acknowledgements

The first author and the second author were supported by the Netherlands Organization for Scientific Research (NWO) grants 639.021.508 and 680-50-0613, respectively, during the period when most of the results were obtained. At present, the first author is partially supported by ERC Advanced Grant Webdam on Foundation of Web data management and the second author is supported by EPSRC grant EP/F002262/1. We would like to thank the anonymous referees of all versions of this paper for their comments and criticism.

Appendix A

A.1. Proof of Lemma 2

- Der1. Use NdAx2, NdAx3, ISAx2.
- Der2. Use NdAx2, NdAx3, ISAx1. As was observed in the main paper, from this moment on we can use all the boolean equivalences for $\neg$ and $\vee$.
- Der3

$$\begin{aligned} A[\phi] &\equiv (A/.)[\phi] && \text{by ISAx5} \\ &\equiv A/.[\phi] && \text{by PrAx4} \end{aligned}$$

- Der4

$$\begin{aligned} A[\mathsf{true}] &\equiv A/.[\mathsf{true}] && \text{by Der3} \\ &\equiv A/. && \text{by PrAx2} \\ &\equiv A && \text{by ISAx5} \end{aligned}$$

- Der5

$$\begin{aligned} A[\mathsf{false}] &\equiv A[\mathsf{false}]/. && \text{by ISAx5} \\ &\equiv \bot && \text{by PrAx1} \end{aligned}$$

- Der6—from NdAx3 and PrAx3.
- Der7

$$\begin{aligned} (A \cup B)[\phi] &\equiv \big((A \cup B)/.\big)[\phi] && \text{by ISAx5} \\ &\equiv (A \cup B)/.[\phi] && \text{by PrAx4} \\ &\equiv A/.[\phi] \cup B/.[\phi] && \text{by ISAx6} \\ &\equiv (A/.)[\phi] \cup (B/.)[\phi] && \text{by PrAx4} \\ &\equiv A[\phi] \cup B[\phi] && \text{by ISAx5} \end{aligned}$$

More often than Der7 itself, we will use the derived property of *monotonicity*:

$$A \leqslant B \quad \text{implies} \quad \begin{cases} A[\phi] \leqslant B[\psi] & \text{for any } \phi, \psi \in \mathsf{NodeEx} \text{ by Der7} \\ \langle A \rangle \leqslant \langle B \rangle & \text{by NdAx3} \end{cases}$$

$$\phi \leqslant \psi \quad \text{implies} \quad A[\phi] \leqslant A[\psi] \quad \text{for any } A \in \mathsf{PathEx} \text{ by PrAx3}$$

- Der8. $.[\neg\langle.\rangle] \leqslant .[\phi]$ is an instance of ISAx7. Monotonicity yields $\langle.[\neg\langle.\rangle]\rangle \leqslant \langle.[\phi]\rangle$. An application of NdAx2 on both sides yields the dual form of Der8.
- Der9. Follows from Der8 by boolean laws.
- Der10

$$\begin{aligned} A &\equiv A[\mathsf{true}] && \text{by Der4} \\ &\equiv A[\phi \vee \neg\phi] && \text{by Der9} \\ &\equiv A[\phi] \cup A[\neg\phi] && \text{by PrAx3} \end{aligned}$$

- Der11

$$\begin{aligned} A &\equiv ./A && \text{by ISAx5} \\ &\equiv (.[\langle A \rangle] \cup .[\neg\langle A \rangle])/A && \text{by Der10} \\ &\equiv .[\langle A \rangle]/A \cup .[\neg\langle A \rangle]/A && \text{by ISAx6} \\ &\equiv .[\langle A \rangle]/A \cup \bot && \text{by PrAx1} \\ &\equiv .[\langle A \rangle]/A && \text{by ISAx7} \end{aligned}$$

- Der12

$$\begin{aligned} \langle A/B \rangle &\equiv \langle A[\langle B \rangle] \rangle && \text{by NdAx4} \\ &\leqslant \langle A[\mathsf{true}] \rangle && \text{monotonicity} \\ &\equiv \langle A \rangle && \text{by Der4} \end{aligned}$$

- Der13—from monotonicity and NdAx2.
- Der14

$$\begin{aligned} \langle A[\phi \wedge \psi] \rangle \wedge \neg\langle A[\psi] \rangle &\leqslant \langle A[\phi \wedge \psi] \rangle \wedge \neg\langle A[\psi \wedge \psi] \rangle && \text{monotonicity} \\ &\equiv \mathsf{false} && \text{by dual of Der9} \end{aligned}$$

- Der15

$$\begin{aligned} \langle A[\phi] \rangle \wedge \neg\langle A[\psi] \rangle &\equiv \langle A[(\phi \wedge \neg\psi) \vee (\phi \wedge \psi)] \rangle \wedge \neg\langle A[\psi] \rangle && \text{boolean} \\ &\equiv (\langle A[\phi \wedge \neg\psi] \rangle \vee \langle A[\phi \wedge \psi] \rangle) \wedge \neg\langle A[\psi] \rangle && \text{by Der6} \\ &\equiv (\langle A[\phi \wedge \neg\psi] \rangle \wedge \neg\langle A[\psi] \rangle) \vee (\langle A[\phi \wedge \psi] \rangle \wedge \neg\langle A[\psi] \rangle) && \text{boolean} \\ &\equiv \langle A[\phi \wedge \neg\psi] \rangle \wedge \neg\langle A[\psi] \rangle && \text{by Der14} \\ &\leqslant \langle A[\phi \wedge \neg\psi] \rangle && \text{boolean} \end{aligned}$$

- Der16

$$\begin{aligned} A[\phi]/.[\neg\phi] &\equiv A[\neg\neg\phi]/.[\neg\phi] && \text{boolean} \\ &\equiv A[\neg\langle.[\neg\phi]\rangle]/.[\neg\phi] && \text{by NdAx2} \\ &\equiv \bot && \text{by PrAx1} \end{aligned}$$

- Der17

$$\begin{aligned} A/\bot &\equiv A[\langle.\rangle]/.[\neg\langle.\rangle] && \text{by Der4} \\ &\equiv \bot && \text{by Der16} \\ \bot/A &\equiv A[\mathsf{false}]/A && \text{by Der5} \\ &\leqslant A[\neg\langle A \rangle]/A && \text{by Der8 and monotonicity} \\ &\equiv \bot && \text{by PrAx2} \end{aligned}$$

Table 9
Additional auxiliary equivalences for full Core XPath.

Der25	$.[\langle \mathsf{s}[\phi]\rangle]$	$\leqslant$	$\mathsf{s}[\phi]/\mathsf{s}^{-1}$	
Der26	$\mathsf{s}/\mathsf{s}^{-1}$	$\equiv$	$.[\langle \mathsf{s}\rangle]$	for $\mathsf{s} \in \{\leftarrow, \rightarrow, \downarrow\}$
Der27	$\uparrow/\downarrow$	$\equiv$	$(\leftarrow^{+} \cup \rightarrow^{+} \cup .)[\langle\uparrow\rangle]$	
Der28	$\langle \mathsf{s}\rangle$	$\equiv$	$\langle \mathsf{s}^{+}\rangle$	
Der29	$\langle \mathsf{s}[\phi]\rangle$	$\equiv$	$\langle \mathsf{s}^{+}[\phi \wedge \neg\langle \mathsf{s}[\phi]\rangle]\rangle$	
Der30	$\langle \mathsf{s}^{*}[\neg\langle \mathsf{s}[\mathsf{true}]\rangle]\rangle$	$\equiv$	true	

– Der18

$$\begin{aligned} A[\phi][\neg\phi] &\equiv A[\phi]/.[\neg\phi] && \text{by Der3} \\ &\equiv \bot && \text{by Der16} \end{aligned}$$

– Der19

$$\begin{aligned} A[\phi \wedge \psi][\neg\phi] &\leqslant A[\phi][\neg\phi] && \text{monotonicity} \\ &\equiv \bot && \text{by Der18} \end{aligned}$$

– Der20

$$\begin{aligned} A[\phi][\psi][\neg\phi \vee \neg\psi] &\equiv A[\phi][\psi][\neg\phi] \cup A[\phi][\psi][\neg\psi] && \text{by PrAx3} \\ &\leqslant A[\phi][\mathsf{true}][\neg\phi] \cup A[\mathsf{true}][\psi][\neg\psi] && \text{by Der8 and monotonicity} \\ &\equiv A[\phi][\neg\phi] \cup A[\psi][\neg\psi] && \text{by Der4} \\ &\equiv \bot && \text{by Der18} \end{aligned}$$

– Der21
First, let us derive

$$\begin{aligned} A[\phi \wedge \psi] &\equiv A[\phi \wedge \psi][\phi] \cup A[\phi \wedge \psi][\neg\phi] && \text{by Der10} \\ &\equiv A[\phi \wedge \psi][\phi] \cup \bot && \text{by Der19} \\ &\equiv A[\phi \wedge \psi][\phi] && \text{by ISAx7} \end{aligned}$$

$A[\phi \wedge \psi] \equiv A[\phi \wedge \psi][\psi]$ is derived analogously. Thus, using monotonicity we get $A[\phi \wedge \psi] \leqslant A[\phi][\psi]$. Conversely,

$$\begin{aligned} A[\phi][\psi] &\equiv A[\phi][\psi][\phi \wedge \psi] \cup A[\phi][\psi][\neg\phi \vee \neg\psi] && \text{by Der10} \\ &\equiv A[\phi][\psi][\phi \wedge \psi] \cup \bot && \text{by Der20} \\ &\equiv A[\phi][\psi][\phi \wedge \psi] && \text{by ISAx7} \end{aligned}$$

Using Der8, monotonicity and Der4 we get that $A[\phi][\psi] \leqslant A[\phi \wedge \psi]$ and Der21 is proved.

– Der22

$$\begin{aligned} A[\phi \wedge \psi] &\equiv A[\phi][\psi] && \text{by Der21} \\ &\equiv A[\phi]/.[\psi] && \text{by Der3} \end{aligned}$$

– Der23—from NdAx2 and Der22.
– Der24—from Der22, commutativity of boolean $\wedge$ and Der11.

A.2. Proof of Theorem 23

We begin by deriving a number of auxiliary references shown in Table 9. We use an additional abbreviation: $\mathsf{s}^{*} = . \cup \mathsf{s}^{+}$.

– Der25. For $\mathsf{s} \in \{\leftarrow, \rightarrow, \downarrow\}$ by TreeAx2 ($\leqslant$ can be even replaced by $\equiv$ then).
For $\mathsf{s} = \uparrow$ by TreeAx3.
– Der26 and Der27—from TreeAx2 and TreeAx3, respectively, using Der4.
– Der28

$$\begin{aligned} \langle \mathsf{s}^{+}\rangle &\equiv \langle \mathsf{s} \cup \mathsf{s}/\mathsf{s}^{+}\rangle && \text{by TreeAx1} \\ &\equiv \langle \mathsf{s}\rangle \vee \langle \mathsf{s}/\mathsf{s}^{+}\rangle && \text{by NdAx3} \\ &\equiv \langle \mathsf{s}\rangle && \text{by Der12} \end{aligned}$$

– Der29

$$
\begin{aligned}
\langle \mathsf{s}[\phi]\rangle &\leqslant \langle \mathsf{s}^+[\phi]\rangle && \text{by LOFT4}\\
&\leqslant \langle \mathsf{s}^+[\phi \wedge \neg\langle \mathsf{s}^+[\phi]\rangle]\rangle && \text{by TransAx1}\\
&\leqslant \langle \mathsf{s}^+[\phi \wedge \neg\langle \mathsf{s}[\phi]\rangle]\rangle && \text{by LOFT4}
\end{aligned}
$$

The use of LOFT4 does not lead to a vicious circle, see its derivation below.

– Der30—follows from Der29 and boolean axioms. Recall that $\mathsf{s}^* = . \cup \mathsf{s}^+$.

Now we can derive the LOFT axioms themselves.

– LOFT0. See the remark on Der2 above.
– LOFT1—already proved, as an instance of Der13 and Der6.
– LOFT2. By boolean reasoning, it is equivalent to

$$\langle \mathsf{s}[\neg\langle \mathsf{s}^{-1}[\phi]\rangle]\rangle \wedge \phi \equiv \mathsf{false}$$

This in turn follows from

$$
\begin{aligned}
\langle \mathsf{s}[\neg\langle \mathsf{s}^{-1}[\phi]\rangle]\rangle \wedge \phi &\equiv \langle .[\langle .[\langle \mathsf{s}[\neg\langle \mathsf{s}^{-1}[\phi]\rangle]\rangle]/.[\phi]\rangle]\rangle && \text{by Der23}\\
&\leqslant \langle .[\langle \mathsf{s}[\neg\langle \mathsf{s}^{-1}[\phi]\rangle]/\mathsf{s}^{-1}/.[\phi]\rangle]\rangle && \text{by Der25}\\
&\equiv \langle .[\langle \mathsf{s}[\neg\langle \mathsf{s}^{-1}[\phi]\rangle]/\mathsf{s}^{-1}[\phi]\rangle]\rangle && \text{by Der3}\\
&\equiv \mathsf{false} && \text{by PrAx1}
\end{aligned}
$$

– LOFT3

$$
\begin{aligned}
\langle \mathsf{s}[\neg\phi]\rangle \wedge \langle \mathsf{s}[\phi]\rangle &\equiv \langle .[\langle \mathsf{s}[\neg\phi]\rangle]/.[\langle \mathsf{s}[\phi]\rangle]\rangle && \text{by Der23}\\
&\leqslant \langle .[\langle \mathsf{s}[\neg\phi]/\mathsf{s}^{-1}/\mathsf{s}[\phi]/\mathsf{s}^{-1}\rangle]\rangle && \text{by Der25}\\
&\equiv \langle .[\langle \mathsf{s}[\neg\phi]/.[\langle \mathsf{s}^{-1}\rangle]/.[\phi]/\mathsf{s}^{-1}\rangle]\rangle && \text{by Der26}\\
&\equiv \langle .[\langle \mathsf{s}[\neg\phi]/.[\phi]/\mathsf{s}^{-1}\rangle]\rangle && \text{by Der11}\\
&\equiv \langle .[\langle \mathsf{s}[\mathsf{false}]/\mathsf{s}^{-1}\rangle]\rangle && \text{by Der22}\\
&\equiv \mathsf{false} && \text{by Der5}
\end{aligned}
$$

– LOFT4

$$
\begin{aligned}
\langle \mathsf{s}^+[\phi]\rangle &\equiv \langle (\mathsf{s} \cup \mathsf{s}/\mathsf{s}^+)[\phi]\rangle && \text{by TreeAx1}\\
&\equiv \langle \mathsf{s}[\phi] \cup (\mathsf{s}/\mathsf{s}^+)[\phi]\rangle && \text{by Der7}\\
&\equiv \langle \mathsf{s}[\phi] \cup \mathsf{s}/\mathsf{s}^+[\phi]\rangle && \text{by PrAx4}\\
&\equiv \langle \mathsf{s}[\phi]\rangle \vee \langle \mathsf{s}/\mathsf{s}^+[\phi]\rangle && \text{by NdAx3}\\
&\equiv \langle \mathsf{s}[\phi]\rangle \vee \langle \mathsf{s}[\langle \mathsf{s}^+[\phi]\rangle]\rangle && \text{by NdAx4}
\end{aligned}
$$

– LOFT5 (see [33]). By boolean reasoning, it boils down to proving that

$$t := \neg\langle \mathsf{s}[\phi]\rangle \wedge \langle \mathsf{s}^+[\phi]\rangle \wedge \neg\langle \mathsf{s}^+[\neg\phi \wedge \langle \mathsf{s}[\phi]\rangle]\rangle \equiv \mathsf{false}$$

This is proven by first observing that

$$
\begin{aligned}
t &\equiv \neg(\langle \mathsf{s}[\phi]\rangle \vee \langle \mathsf{s}[\neg\phi \wedge \langle \mathsf{s}[\phi]\rangle]\rangle) \wedge \langle \mathsf{s}/\mathsf{s}^+[\phi]\rangle \wedge \neg\langle \mathsf{s}/\mathsf{s}^+[\neg\phi \wedge \langle \mathsf{s}[\phi]\rangle]\rangle && \text{by LOFT4}\\
&\equiv \neg\langle \mathsf{s}[\phi \vee (\neg\phi \wedge \langle \mathsf{s}[\phi]\rangle)]\rangle \wedge \langle \mathsf{s}/\mathsf{s}^+[\phi]\rangle \wedge \neg\langle \mathsf{s}/\mathsf{s}^+[\neg\phi \wedge \langle \mathsf{s}[\phi]\rangle]\rangle && \text{by Der6}\\
&\equiv \neg\langle \mathsf{s}[\phi \vee \langle \mathsf{s}[\phi]\rangle]\rangle \wedge \langle \mathsf{s}/\mathsf{s}^+[\phi]\rangle \wedge \neg\langle \mathsf{s}/\mathsf{s}^+[\neg\phi \wedge \langle \mathsf{s}[\phi]\rangle]\rangle && \text{boolean}\\
&\equiv \neg\langle \mathsf{s}[\phi \vee \langle \mathsf{s}[\phi]\rangle]\rangle \wedge \langle \mathsf{s}[\langle \mathsf{s}^+[\phi]\rangle]\rangle \wedge \neg\langle \mathsf{s}[\langle \mathsf{s}^+[\neg\phi \wedge \langle \mathsf{s}[\phi]\rangle]\rangle]\rangle && \text{by NdAx4}\\
&\leqslant \langle \mathsf{s}[\neg(\phi \vee \langle \mathsf{s}[\phi]\rangle) \wedge \langle \mathsf{s}^+[\phi]\rangle \wedge \neg\langle \mathsf{s}^+[\neg\phi \wedge \langle \mathsf{s}[\phi]\rangle]\rangle]\rangle && \text{by Der15}\\
&\leqslant \langle \mathsf{s}[\neg\langle \mathsf{s}[\phi]\rangle \wedge \langle \mathsf{s}^+[\phi]\rangle \wedge \neg\langle \mathsf{s}^+[\neg\phi \wedge \langle \mathsf{s}[\phi]\rangle]\rangle]\rangle && \text{monotonicity}\\
&= \langle \mathsf{s}[t]\rangle
\end{aligned}
$$

Thus, we get

$$\begin{aligned} t &\leqslant \langle \mathsf{s}[t] \rangle && \text{by the above} \\ &\leqslant \langle \mathsf{s}^+[t \wedge \neg\langle \mathsf{s}[t]\rangle] \rangle && \text{Der29} \\ &\equiv \langle \mathsf{s}[\mathsf{false}] \rangle && \text{by the above} \\ &\equiv \mathsf{false} && \text{by Der5} \end{aligned}$$

- LOFT6—already proved, as an instance of Der29.
- LOFT7 does not need to be proved, being an instance of an axiom.
- LOFT8. By boolean reasoning, it is equivalent to

$$\langle \downarrow[\neg\langle\leftarrow\rangle \wedge \neg\langle \rightarrow^*[\phi]\rangle] \rangle \wedge \langle \downarrow[\phi] \rangle \equiv \mathsf{false}$$

$$\begin{aligned} \langle \downarrow[\neg\langle\leftarrow\rangle \wedge \neg\langle \rightarrow^*[\phi]\rangle] \rangle \wedge \langle \downarrow[\phi] \rangle &\equiv \langle .[\langle \downarrow[\neg\langle\leftarrow\rangle \wedge \neg\langle \rightarrow^*[\phi]\rangle]\rangle]/.[\langle \downarrow[\phi]\rangle] \rangle && \text{by Der23} \\ &\equiv \langle \downarrow[\neg\langle\leftarrow\rangle \wedge \neg\langle \rightarrow^*[\phi]\rangle]/\uparrow/\downarrow[\phi]/\uparrow \rangle && \text{by Der26} \\ &\equiv \langle \downarrow[\neg\langle\leftarrow\rangle \wedge \neg\langle \rightarrow^*[\phi]\rangle]/(\leftarrow^+ \cup \rightarrow^+ \cup .)[\langle\uparrow\rangle][\phi]/\uparrow \rangle && \text{by Der27} \\ &\equiv \langle \downarrow[\neg\langle\leftarrow\rangle \wedge \neg\langle \rightarrow^*[\phi]\rangle]/(\leftarrow^+ \cup \rightarrow^+ \cup .)[\phi]/\uparrow \rangle && \text{by Der11} \\ &\leqslant \langle (\downarrow[\neg\langle\leftarrow\rangle]/\leftarrow^+/\uparrow) \cup (\downarrow[\neg\langle \rightarrow^*[\phi]\rangle]/\rightarrow^*[\phi]/\uparrow) \rangle && \text{by ISAx6} \\ &\equiv \mathsf{false} && \text{by PrAx1} \end{aligned}$$

- LOFT9

$$\begin{aligned} \langle \downarrow[\phi] \rangle &\leqslant \langle \downarrow[\mathsf{true}] \rangle && \text{monotonicity} \\ &\equiv \langle \downarrow[\langle \leftarrow^*[\neg\langle\leftarrow[\mathsf{true}]\rangle]\rangle] \rangle && \text{by Der30} \\ &\equiv \langle \downarrow[\neg\langle\leftarrow\rangle] \rangle \vee \langle \downarrow[\langle \leftarrow^+[\neg\langle\leftarrow\rangle]\rangle] \rangle && \text{by Der6} \end{aligned}$$

But now

$$\begin{aligned} \langle \downarrow[\langle \leftarrow^+[\neg\langle\leftarrow\rangle]\rangle] \rangle &\equiv \langle \downarrow[\langle \leftarrow^+[\uparrow][\neg\langle\leftarrow\rangle]\rangle] \rangle && \text{by TreeAx4} \\ &\leqslant \langle \downarrow[\langle \uparrow/\downarrow[\neg\langle\leftarrow\rangle]\rangle] \rangle && \text{by Der27} \\ &\equiv \langle \downarrow/\uparrow/\downarrow[\neg\langle\leftarrow\rangle] \rangle && \text{by NdAx4} \\ &\equiv \langle .[\downarrow]/\downarrow[\neg\langle\leftarrow\rangle] \rangle && \text{by Der26} \\ &\equiv \langle \downarrow[\neg\langle\leftarrow\rangle] \rangle && \text{by Der11} \end{aligned}$$

Thus, we got $\langle \downarrow[\phi] \rangle \leqslant \langle \downarrow[\neg\langle\leftarrow\rangle] \rangle$.
The proof of $\langle \downarrow[\phi] \rangle \leqslant \langle \downarrow[\neg\langle\rightarrow\rangle] \rangle$ is analogous.

- LOFT10. First observe that by boolean reasoning, it is equivalent to

$$\langle\leftarrow\rangle \vee \langle\rightarrow\rangle \leqslant \langle\uparrow\rangle$$

$$\begin{aligned} \langle\leftarrow\rangle \vee \langle\rightarrow\rangle &\equiv \langle\leftarrow^+\rangle \vee \langle\rightarrow^+\rangle && \text{by Der28} \\ &\equiv \langle \leftarrow^+ \cup \rightarrow^+ \rangle && \text{by NdAx3} \\ &\equiv \langle \leftarrow^+[\langle\uparrow\rangle] \cup \rightarrow^+[\langle\uparrow\rangle] \rangle && \text{by TreeAx4} \\ &\equiv \langle (\leftarrow^+ \cup \rightarrow^+)[\langle\uparrow\rangle] \rangle && \text{by Der7} \\ &\leqslant \langle \uparrow/\downarrow \rangle && \text{by Der27} \\ &\leqslant \langle\uparrow\rangle && \text{by Der12} \end{aligned}$$

References

[1] A. Balmin, F. Özcan, A. Singh, E. Ting, Grouping and optimization of XPath expressions in DB2® pure XML, in: SIGMOD '08: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, New York, NY, USA, 2008, ACM, 2008, pp. 1065–1074.
[2] M. Benedikt, W. Fan, G. Kuper, Structural properties of XPath fragments, Theoretical Computer Science 336 (1) (2005) 3–31.
[3] P. Blackburn, M. de Rijke, Y. Venema, Modal Logic, Cambridge University Press, 2001.
[4] P. Blackburn, W. Meyer-Viol, M. de Rijke, A proof system for finite trees, in: H.K. Büning (Ed.), Computer Science Logic, in: Lecture Notes in Computer Science, vol. 1092, Springer, 1996, pp. 86–105.
[5] Z. Ésik, W. Kuich, A semiring-semimodule generalization of omega-context-free languages, in: J. Karhumäki, H.A. Maurer, G. Paun, G. Rozenberg (Eds.), Theory Is Forever, in: Lecture Notes in Computer Science, vol. 3113, Springer, 2004, pp. 68–80.
[6] K. Fine, Normal forms in modal logic, Notre Dame Journal of Formal Logic 16 (2) (1975) 229–234.
[7] R. Goldblatt, Logics of Time and Computation, Lecture Notes, vol. 7, CSLI Publications, 1987.
[8] G. Gottlob, C. Koch, Monadic queries over tree-structured data, in: Proceedings of LICS'02, Los Alamitos, California, 2002, IEEE Computer Society Press, 2002, pp. 189–202.
[9] G. Gottlob, C. Koch, R. Pichler, Efficient algorithms for processing XPath queries, ACM Transactions on Database Systems 30 (2) (2005) 444–491.
[10] T. Hillenbrand, A. Buch, R. Vogt, B. Löchner, Waldmeister: High performance equational deduction, Journal of Automated Reasoning 18 (2) (1997) 265–270.
[11] R. Hirsch, Peirce algebras and boolean modules, Journal of Logic and Computation 17 (2) (2007) 255–283.
[12] M. Hollenberg, An equational axiomatization of dynamic negation and relational composition, Journal of Logic, Language and Information 6 (4) (1997) 381–401.
[13] E.V. Huntington, Boolean algebra. A correction, Transactions of AMS 35 (2) (1933) 557–558.
[14] E.V. Huntington, New sets of independent postulates for the algebra of logic with special reference to Whitehead and Russell's "Principia Mathematica", Transactions of AMS 35 (1) (1933) 274–304.
[15] P. Jipsen, Computer-aided investigations of relation algebras, PhD thesis, Mathematics Department, Vanderbilt University, 1992.
[16] S. Kleene, Representation of events in nerve nets and finite automata, in: C. Shannon, J. McCarthy (Eds.), Automata Studies, Princeton University Press, Princeton, NJ, 1956, pp. 3–42.
[17] D. Kozen, A completeness theorem for Kleene algebras and the algebra of regular events, Information and Computation 110 (2) (1994) 366–390.
[18] M. Krishnaprasad, Z.H. Liu, A. Manikutty, J.W. Warner, V. Arora, S. Kotsovolos, Query rewrite for XML in Oracle XML DB, in: VLDB '04: Proceedings of the Thirtieth International Conference on Very Large Data Bases, VLDB Endowment, 2004, pp. 1134–1145.
[19] M. Marx, XPath with conditional axis relations, in: E. Bertino, S. Christodoulakis, D. Plexousakis, V. Christophides, M. Koubarakis, K. Böhm, E. Ferrari (Eds.), Proceedings of the 9th International Conference on Extending Database Technology, EDBT 2004, in: Lecture Notes in Computer Science, vol. 2992, Springer, 2004.
[20] M. Marx, M. de Rijke, Semantic characterizations of navigational XPath, SIGMOD Rec. 34 (2) (2005) 41–46.
[21] W. McCune, Prover9 and Mace4 website, http://www.cs.unm.edu/~mccune/.
[22] W. McCune, Solution of the Robbins problem, Journal of Automated Reasoning 19 (3) (1997) 263–276.
[23] L.S. Moss, Finite models constructed from canonical formulas, Journal of Philosophical Logic 36 (6) (2007) 605–640.
[24] V. Pratt, Dynamic algebras: Examples, constructions, applications, Studia Logica 50 (3–4) (1991) 571–605.
[25] M. Reynolds, An axiomatization for Until and Since over the reals without the IRR rule, Studia Logica 51 (2) (1992) 165–193.
[26] A. Tarski, On the calculus of relations, Journal of Symbolic Logic 6 (3) (1941) 73–89.
[27] A. Tarski, S. Givant, A Formalization of Set Theory without Variables, AMS Colloquium Publications, vol. 41, Providence, RI, 1987.
[28] B. ten Cate, G. Fontaine, T. Litak, Some modal aspects of XPath, Earlier version published in Preliminary Proceedings of Methods for Modalities 2007. Extended journal version in preparation. See http://www.dcs.bbk.ac.uk/~tadeusz/papers/, for the most recent version.
[29] B. ten Cate, C. Lutz, The complexity of query containment in expressive fragments of XPath 2.0, in: Proceedings of PODS'07, ACM, New York, NY, USA, 2007, pp. 73–82.
[30] B. ten Cate, M. Marx, Navigational XPath: Calculus and algebra, SIGMOD Rec. 36 (2) (2007) 19–26.
[31] B. ten Cate, M. Marx, Axiomatizing the logical core of XPath 2.0, Theory of Computing Systems 44 (4) (2009) 561–589, Open access: http://www.springerlink.com/content/m62011j670270282/fulltext.pdf.
[32] J. van Benthem, Exploring Logical Dynamics. Studies in Logic, Language and Information, CSLI Publications, Stanford, 1996.
[33] J. van Benthem, Modal frame correspondence and fixed-points, Studia Logica 83 (1) (2006) 133–155.
[34] Y. Venema, Completeness via completeness: Since and until, in: M. de Rijke (Ed.), Diamonds and Defaults, in: Synthese Library, vol. 229, Kluwer Academic Publishers, 1993, pp. 279–286.
[35] Y. Venema, A crash course in arrow logic, in: Arrow Logic and Multi-Modal Logic, Center for the Study of Language and Information, Stanford, CA, USA, 1997, pp. 3–34.
[36] W3C, XML path language (XPath): Version 1.0. W3C recommendation, http://www.w3.org/TR/xpath.html.