Saturation, Definability, and Separation for XPath on Data Trees

Sergio Abriola[1], María Emilia Descotte[1], and Santiago Figueira[1,2]

[1] University of Buenos Aires, Argentina
[2] CONICET, Argentina

Abstract. We study the expressive power of some fragments of XPath equipped with (in)equality tests over data trees.
Our main results are the definability theorems, which give necessary and sufficient conditions under which a class of data trees can be defined by a node expression or set of node expressions, and our separation theorems, which give sufficient conditions under which two disjoint classes of data trees can be separated by a class of data trees definable in XPath.

Keywords: XPath · data tree · bisimulation · definability · first-order logic · ultraproduct · saturation · separation.

1 Introduction

The abstraction of an XML document is a data tree, i.e. a tree whose every node contains a tag or label (such as *LastName*) from a finite domain, and a data value (such as *Smith*) from an infinite domain. XPath is the most widely used query language for XML documents; it is an open standard and constitutes a World Wide Web Consortium (W3C) Recommendation [3]. XPath$_=$ has syntactic operators to navigate the tree using the 'child', 'parent', 'sibling', etc. accessibility relations, and can make tests on intermediate nodes. It can express properties of the underlying tree structure of the XML document, such as "*the root of the tree has a child labeled a and a child labeled b*", and it can express conditions on the actual data contained in the attributes, such as "*the root of the tree has two children with same tag a but different data value*".

First, we provide notions of saturation and ultraproducts that are adequate for XPath$_=$, and show that bisimulation coincides with logical equivalence over saturated data trees. Using these tools, we show definability theorems, giving necessary and sufficient conditions under which a class of data trees can be defined by a node expression or set of node expressions of XPath$_=$. Finally we give separation results, providing sufficient conditions under which two disjoint classes of data trees can be separated by a class of data trees definable in XPath$_=$.

While on this work we will only show results for the fragment of XPath that can only navigate via the 'child' accessibility relation, similar results hold for the vertical fragment having both the 'child' and 'parent' navigational operators.

The results on definability of this paper appeared originally in [1].

2 Preliminaries

Data trees. We say that $\mathcal{T}$ is a **data tree** if it is a tree from $\mathit{Trees}(\mathbb{A} \times \mathbb{D})$, where $\mathbb{A}$ is a finite set of **labels** and $\mathbb{D}$ is an infinite set of **data values**. The data of a node x is denoted $\mathit{data}(x)$, and its label is $\mathit{label}(x)$. The set of nodes of a data tree $\mathcal{T}$ is denoted T.

Downward XPath with data tests. We consider a fragment of XPath that corresponds to the navigational part of XPath 1.0 with data equality and inequality. $\text{XPath}_=$ is a two-sorted language, with **path expressions** (that we write α, β, γ) and **node expressions** (that we write φ, ψ, η). The **downward XPath**, notated $\text{XPath}^{\downarrow}_{=}$ is defined by mutual recursion as follows:

$$\begin{array}{lll} \alpha, \beta & ::= & o \mid [\varphi] \mid \alpha\beta \mid \alpha \cup \beta \qquad\qquad o \in \{\varepsilon, \downarrow\} \\ \varphi, \psi & ::= & a \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \langle\alpha\rangle \mid \langle\alpha = \beta\rangle \mid \langle\alpha \neq \beta\rangle \qquad a \in \mathbb{A} \end{array}$$

Node expressions represent properties on nodes. They are evaluated in nodes, and, intuitively, $\langle\alpha = \beta\rangle$ is true at x if there are two paths starting in x, one satisfying the property α, and the other satisfying the property β, which end in nodes with equal data value. On the other hand, path expressions represent properties on paths. They are evaluated in pairs of nodes. For instance $\downarrow$ is true at (x, y) if y is a child of x, and $[\varphi]$ is true at x, y if $x = y$ and x satisfies φ.

Let $\mathcal{T}$ and $\mathcal{T}'$ be data trees, and let $u \in T$, $u' \in T'$. We say that $\mathcal{T}, u$ and $\mathcal{T}', u'$ are **logically equivalent for** $\text{XPath}^{\downarrow}_{=}$ if no $\text{XPath}^{\downarrow}_{=}$ can distinguish node u from u'.

Bisimulations. Notions of bisimulation present a way to determine whether two pointed data trees can be distinguished by a series of moves in XPath. We do not reproduce them here, but it is worth mentioning that they are forms of back-and-forth conditions over two data trees.

The main previous result in the literature establishing the connection between bisimulation and equivalence is the following:

Theorem 1. *[4] If $\mathcal{T}, u$ is bisimilar to $\mathcal{T}', u'$, then they are logically equivalent. If $\mathcal{T}$ and $\mathcal{T}'$ are finitely branching, the other implication also holds.*

3 Saturation and quasi-ultraproducts

We introduce a notion of saturation for the downward fragment of XPath, and show that the reverse implication of Theorem 1 is true over saturated data trees. Saturation is the key ingredient to show the Definability theorems, but their use lays hidden in the proof.

Saturation. Let $\langle \Sigma_1, \ldots, \Sigma_n \rangle$ and $\langle \Gamma_1, \ldots, \Gamma_m \rangle$ be tuples of sets of $\text{XPath}^{\downarrow}_{=}$-formulas. Given a data tree $\mathcal{T}$ and $u \in T$, we say that $\langle \Sigma_1, \ldots, \Sigma_n \rangle$ and $\langle \Gamma_1, \ldots, \Gamma_m \rangle$ are $=^{\downarrow}_{n,m}$**-satisfiable** [resp. $\neq^{\downarrow}_{n,m}$**-satisfiable**] at $\mathcal{T}, u$ if there exist $v_0 \to v_1 \to \cdots \to v_n \in T$ and $w_0 \to w_1 \to \cdots \to w_m \in T$ such that $u = v_0 = w_0$ and

1. for all $i \in \{1, \ldots, n\}$, $\mathcal{T}, v_i \models \Sigma_i$;
2. for all $j \in \{1, \ldots, m\}$, $\mathcal{T}, w_j \models \Gamma_j$; and
3. $data(v_n) = data(w_m)$ [resp. $data(v_n) \neq data(w_m)$].

We say that $\langle \Sigma_1, \ldots, \Sigma_n \rangle$ and $\langle \Gamma_1, \ldots, \Gamma_m \rangle$ are $=^{\downarrow}_{n,m}$**-finitely satisfiable** [resp. $\neq^{\downarrow}_{n,m}$**-finitely satisfiable**] at $\mathcal{T}, u$ if for every finite $\Sigma'_i \subseteq \Sigma_i$ and finite $\Gamma'_j \subseteq \Gamma_j$, we have that $\langle \Sigma'_1, \ldots, \Sigma'_n \rangle$ and $\langle \Gamma'_1, \ldots, \Gamma'_m \rangle$ are $=^{\downarrow}_{n,m}$-satisfiable [resp. $\neq^{\downarrow}_{n,m}$-satisfiable] at $\mathcal{T}, u$.

Definition 2. *We say that a data tree $\mathcal{T}$ is* $\downarrow$***-saturated*** *if for every $n, m \in \mathbb{N}$, every pair of tuples $\langle \Sigma_1, \ldots, \Sigma_n \rangle$ and $\langle \Gamma_1, \ldots, \Gamma_m \rangle$ of sets of $XPath^{\downarrow}_{=}$-formulas, every $u \in T$, and $\star \in \{=, \neq\}$, the following is true:*

if $\langle \Sigma_1, \ldots, \Sigma_n \rangle$ and $\langle \Gamma_1, \ldots, \Gamma_m \rangle$ are $\star^{\downarrow}_{n,m}$-finitely satisfiable at $\mathcal{T}, u$ then $\langle \Sigma_1, \ldots, \Sigma_n \rangle$ and $\langle \Gamma_1, \ldots, \Gamma_m \rangle$ are $\star^{\downarrow}_{n,m}$-satisfiable at $\mathcal{T}, u$.

Proposition 3. *For $\downarrow$-saturated data trees, bisimulation coincides with logical equivalence.*

Quasi-ultraproducts We introduce the notion of quasi-ultraproduct, a variant of the usual notion of first-order model theory, which will be needed for the definability theorems. Some of our results for quasi-ultraproducts make use of the fundamental theorem of ultraproducts (see e.g. [2, Thm. 4.1.9]).

Definition 4. *Suppose $(\mathcal{T}_i, u_i)_{i \in I}$ is a family of pointed data trees, U is an ultrafilter over I, $\mathcal{T}^*$ is the ultraproduct of $(\mathcal{T}_i, u_i)_{i \in I}$, and u^* is the ultralimit of $(u_i)_{i \in I}$. The* $\downarrow$***-quasi ultraproduct*** *of $(\mathcal{T}_i, u_i)_{i \in I}$ modulo U is the pointed data tree $(\mathcal{T}^*|u^*, u^*)$, where $\mathcal{T}^*|u^*$ denotes the subtree of $\mathcal{T}^*$ induced by al the descendants of u^*. As a particular case one has the notion of* $\downarrow$***-quasi ultrapower****.*

4 Definability

Definability theorems address the question of which properties of models can be defined via formulas of the logic. If K is a class of pointed data trees, we denote its complement by $\overline{K}$.

Theorem 5. *Let K be a class of pointed data trees. Then K is definable by a set of $XPath^{\downarrow}_{=}$-formulas iff K is closed under $\downarrow$-bisimulations and $\downarrow$-quasi ultraproducts, and $\overline{K}$ is closed under $\downarrow$-quasi ultrapowers.*

Theorem 6. *Let K be a class of pointed data trees. Then K is definable by an $XPath^{\downarrow}_{=}$-formula iff both K and $\overline{K}$ are closed under $\downarrow$-bisimulations and $\downarrow$-quasi ultraproducts.*

The notion of ℓ-bisimulation is a restricted version of $\downarrow$-bisimulations. It has been shown to coincide with the notion of ℓ-equivalence, which informally means *indistinguishable by $XPath^{\downarrow}_{=}$ formulas that cannot "see" beyond ℓ 'child'-steps from the current point of evaluation.*

Theorem 7. *Let K be a class of pointed data trees. Then K is definable by a formula of $XPath^{\downarrow}_{=}$ iff K is closed by ℓ-bisimulations for $XPath^{\downarrow}_{=}$ for some ℓ.*

5 Separation

Separation theorem provide conditions under which two disjoint classes of pointed models can be separated by a class definable in the logic.

Theorem 8. *Let K_1 and K_2 be two disjoint classes of pointed data trees such that K_1 is closed under $\downarrow$-bisimulations and $\downarrow$-quasi ultraproducts and K_2 is closed under $\downarrow$-bisimulations and $\downarrow$-quasi ultrapowers. Then there exists a third class K which is definable by a set of $XPath^{\downarrow}_{=}$-formulas, contains K_1 and is disjoint from K_2.*

Theorem 9. *Let K_1 and K_2 be two disjoint classes of pointed data trees closed under $\downarrow$-bisimulations and $\downarrow$-quasi ultraproducts. Then there exists a third class K which is definable by an $XPath^{\downarrow}_{=}$-formula, contains K_1 and is disjoint from K_2.*

References

1. Sergio Abriola, María Emilia Descotte, and Santiago Figueira. Definability for downward and vertical Xpath on data trees. In *Logic, Language, Information, and Computation - 21st International Workshop, WoLLIC 2014, Valparaíso, Chile, September 1-4, 2014. Proceedings*, pages 20–35, 2014.
2. C.C. Chang and H.J. Keisler. *Model theory*. Studies in logic and the foundations of mathematics. North-Holland, 1990.
3. J. Clark and S. DeRose. XML path language (XPath). Website, 1999. W3C Recommendation. `http://www.w3.org/TR/xpath`.
4. D. Figueira, S. Figueira, and C. Areces. Basic model theory of XPath on data trees. In *ICDT*, pages 50–60, 2014.