ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.17, No.5, May 2006, pp.1089−1097
DOI: 10.1360/jos171089

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种 XML 的模型论语义*

刘升平, 林作铨+, 梅 婧, 岳安步

(北京大学 信息科学系,北京 100871)

A Model-Theoretic Semantics for XML

LIU Sheng-Ping, LIN Zuo-Quan+, MEI Jing, YUE An-Bu

(Department of Information Science, Peking University, Beijing 100871, China)

+ Corresponding author: Phn: +86-10-62757175, E-mail: lz@is.pku.edu.cn, http://www.is.pku.edu.cn

Liu SP, Lin ZQ, Mei J, Yue AB. A model-theoretic semantics for XML. *Journal of Software*, 2006,17(5): 1089−1097. http://www.jos.org.cn/1000-9825/17/1089.htm

Abstract: The problem that XML formally governs syntax only but not semantics has been recognized as a serious barrier for XML-Based data integration and the extension of current Web to the semantic Web. To address this problem, the XML Semantics Definition Language (XSDL) is proposed to explicitly express the XML author's intended meaning and a model-theoretic semantics for XML. In this way, the XML becomes a sub-language of RDF (resource description framework) in expressivity and the XML data can be semantics-preserving transformed to the RDF data. The semantic validity and entailment problem of XML documents are further provided and they are reduced to the knowledge base unsatisfiability problem in description logic language $\mathcal{SHOIN}(\mathcal{D})$.

Key words: XML; semantics; semantic Web

摘 要: XML 只能表示语法而不能表达形式化语义,这个问题导致 XML 数据集成以及扩展当前 Web 到语义 Web 非常困难.为了解决该问题,提出了一种 XML 语义定义语言 XSDL,让 XML 文档作者清晰地表达 XML 文档中的语义信息,并提出了一种 XML 的模型论语义.这样,XML 成为一种表达能力比资源描述框架(resource description framework,简称 RDF)稍弱的 Web 知识表示语言,且 XML 数据可以保留语义转换到 RDF 数据.此外,还提出了 XML 文档的语义有效性和 XML 文档的推理问题,并把它们规约到描述逻辑语言$\mathcal{SHOIN}(\mathcal{D})$的知识库不可满足性问题.

关键词: XML;语义;语义 Web
中图法分类号: TP393 文献标识码: A

可扩展标记语言 XML 作为 Web 上数据表示和交换的标准,已经获得了巨大的成功.但是,XML 的一个重要缺陷也越来越为人们所认识,即 XML 只能表达数据的语法,而不能表达形式化的语义.XML 的一个基本思想

* Supported by the National Natural Science Foundation of China under Grant Nos.60373002, 60496322 (国家自然科学基金); the NKBRPC under Grant No.2004CB318000 (国家重点基础研究发展规划(973))

Received 2004-08-05; Accepted 2005-10-27

是,XML 文档中的数据是通过标签,以一种有意义的和自描述的方式来描述的,且标签的名字是领域专家精心选取的[1].这些标签体现了人们的共识.例如,标签⟨price⟩对人来说意为价格,这样就可以推断出标签中包含的数据就是关于价格的.但 XML 本身,包括 DTD 或 XML Schema,都没有提供形式化的机制来说明标签到底是什么含义.因此,XML 处理器无法理解 XML 文档中标签的含义.对它来说,标签⟨price⟩与 HTML 标签⟨H1⟩的含义是没有什么区别的.所以说 XML 只表示语法,而不表达形式化语义.可是,XML 具有隐式的语义,因为 XML 文档所包含的语义信息没有被显式地表达出来,而是隐藏在人们的头脑中.XML 文档中蕴含的语义信息称为 XML 语义.

由于 XML 的隐式语义,开发人员在开发 XML 应用程序的时候,需要预先知道标签和文档结构的含义,并把这种含义硬编码在程序中[2],如⟨price⟩标签中的数据就是表示物品的价格.如果 XML 文档有多方使用者,由于没有规范说明标签的含义,他们可能会对标签产生不同的理解,且把各自对标签的理解硬编码在程序中,从而导致应用程序难以互操作,这是 XML 数据集成和应用集成所要克服的困难.

XML 的隐式语义也使得它不适合表示数据的语义信息.因此,作为下一代的 Web,即语义 Web[3],需要提出一种新的元数据表示语言 RDF[4],用以表达 Web 上内容的语义信息.语义 Web 引入了新的基于知识表示的技术,如 RDF Schema 和 Web 本体定义语言 OWL,用以定义领域词汇及内容模型.自从剑桥公报[5]以来,XML 和 RDF 领域都认识到对 XML 和 RDF 关系的理解,以及从 XML 数据映射到具有语义的 RDF 数据是一个非常重要的课题,到目前为止并没有得到很好的解决[6].其原因之一就是 XML 和 RDF 之间存在巨大的差异性:XML 只能表示语法,而 RDF 具有语义.尽管 RDF 的语法是基于 XML 的,但其数据模型是基于知识表示常用的边标识的有向图模型,而 XML 的数据模型是节点标识的有序的树.这样,现存 Web 大量的 XML 数据难以直接转换为语义 Web 上需要的具有语义描述的数据,语义 Web 也难以作为当前 Web 的直接扩展.这也会阻碍工业界的广泛接受.

为了 XML 获得更为广泛的应用和语义 Web 的发展,XML 数据的语义需要形式化的表达,以便使 XML 语义能够被计算机理解.Patel-Schneider 和 Simeon[9]提出了一种 XML 的模型论语义,并提出了阴阳 Web 理论,统一了 XML 和 RDF 的语义.其基本思想是,把 RDF 文档也看成 XML 文档,并给 XML 一个和 RDF 语义兼容的模型论语义.在其语义中,元素节点被解释为以这个节点名为名称的类的实例,父子节点以及属性节点和它的值都解释为关系.但是,由于 XML 表达的语义信息隐含在 XML 文档的标签和结构中,并没有规范来约束 XML 如何表达语义信息,因此,XML 表达语义的方式是多种多样的,如一个元素节点可能表示一个对象,但也可能表示一个属性.另外,一个元素节点还可能只有在符合一定条件下才表达某个对象.这样,如果直接给 XML 文档一个模型论语义,则这个语义将很难符合 XML 文档作者的本意,因此限制了它在实际中的应用.

我们需要有一种规范语言,使得用户能够自己用它去准确地描述 XML 文档中蕴涵的语义信息.MDL (meaning definition language)[8]是一个由工业界提出的表达 XML 语义的语言,它用 XPath 表达式选取 XML 中的节点,并把这些节点映射到 UML 的对象、属性和关联.但由于它采用 UML 作为 XML 的语义模型,因此,并没有形式化 XML 语义,也不能支持对 XML 文档的推理,难以弥补 XML 和语义 Web 之间的差距.

我们提出了 XML 语义定义语言 XSDL(XML semantics definition language),其基本思想是,用 OWL DL 表达 XML 的语义信息,用 XPath 表达式选取含有语义信息的节点集合,再定义一个映射,把 XML 映射到 OWL DL 的本体.我们给出了一种结合 XSDL 的 XML 的模型论语义,其基本做法是:首先定义 XML 的简单解释,即给 XML 文档一个初步的解释,例如,XML 中的一些元素节点被解释为论域中的实例,但具体属于哪个类是没有限制的;然后定义 XML 的 XSDL 解释,即结合 XSDL 定义,给 XML 文档一个符合作者本意的准确解释,例如,一些元素节点进一步解释为某个类的实例(根据 XSDL 定义).基于 XSDL 的 XML 模型论语义,XML 可看成一种表达能力比 RDF 稍弱的 Web 知识表示语言,XML 数据可以保留语义地转换为 RDF 数据.进一步地,我们引入 XML 文档的语义有效性,可以用来检查 XML 是否满足一些 XML Schema 难以表达的语义完整性约束,并引入 XML 文档推理问题,讨论了 XML 文档之间的语义蕴含关系.我们把 XML 文档的语义有效性检查和语义蕴含推理都规约到描述逻辑语言$\mathcal{SHOIN}\boxplus\Delta\boxtimes$ [10]上的推理问题,因此,可借用描述逻辑的推理机实现.

本文第 1 节首先分析 XML 是如何隐式地表示语义.第 2 节提出 XML 语义定义语言 XSDL.第 3 节给出结

合 XSDL 的 XML 模型论语义.第 4 节讨论相关工作.最后,总结本文的结果并指出下一步的工作.

1 XML 语义

我们知道,XML 语义是 XML 文档中蕴含的语义信息.XML 是设计作为统一的数据格式,并不适合表达领域知识中的概念模型(包括概念、属性以及它们之间的关系).但是,当用户把数据表达成 XML 文档的时候,数据的语义信息其实已经隐含在 XML 文档的标签和结构中了,虽然这些语义信息并不能被机器所理解.在 XML 中,能够表达概念、属性和关系的成分是元素节点、属性节点、节点的嵌套、节点的序和交叉引用(ID/IDREF)等.

例 1:关于 XML 隐式语义的 XML 片断.

```
〈wineMerchant name=“Bristol Bottlers”〉
    〈wine id=“w100”〉
        〈name〉Vielles Bottes〈/name〉  〈colour〉black〈/colour〉
    〈/wine〉
〈/wineMerchant〉
```

上例中隐含丰富的语义信息:有一个名为 Bristol Bottlers 的酒商,销售了一种黑色的名为 Vielles Bottes 的酒,其中 wineMerchant 节点和 wine 节点分别表示了一个酒商和酒的实例,它们的嵌套表示了一种销售关系.

但由于没有特定的规范去指导如何在 XML 中隐式地表达语义信息,用户几乎可以用任意的方式去表达语义信息,例如:

```
〈SoldOn date=“20040722”〉
    〈pen code=“SKU001” price=“12.75” units=“1”/〉
    〈pencil code=“SKU002” price =“2.45” units=“10”/〉
〈/SoldOn〉
```

其中,SoldOn 并不表示一个实例或者类型,它用于根据相同的属性值对一些元素进行分组.此外,节点代表的含义可能取决于一些上下文节点以及满足一些条件,有时 XML 中的文档序也蕴涵了重要的含义.

总之,由于 XML 隐式语义的表达方式的多样性,用户在设计 XML 文档时,并不一定会遵循知识表示的思想,这使得自动从 XML 文档中获取其语义非常困难.

2 XML 语义定义语言 XSDL

仅凭对 XML 的语法分析难以提取出 XML 中的语义信息.我们需要一种 XML 语义的定义语言,让用户用它把 XML 语义清晰地表达出来.这个语言至少应包含两部分内容:1) 一个形式化语言,用以表达 XML 的语义信息;2) 一个映射语言,把 XML 中能表达语义的成分映射到这个形式语言.在 XSDL 中,形式语言采用 OWL DL,因为 OWL 已经成为 Web 本体语言的标准,且 OWL DL 具有与一阶逻辑相容的语义,还有较为丰富的表达能力和可判定的推理能力;映射语言基于模式附件框架 SAF(schema adjuncts framework[9]).SAF 是一种常用的用于描述作用在 XML Schema 层次上的应用相关信息的框架.

下面,我们首先介绍 XSDL 文档的基本结构,然后说明怎样用 XSDL 表示 XML 中多样的语义信息,以及 XSDL 的抽象语法.限于篇幅,本文只给出 XSDL 中一些有代表性的功能.

2.1 基于SAF的文档结构

DTD 或 XML Schema 描述了文档的结构模型,而 SAF 则进一步提供一种描述关于 Schema 的元数据的框架,如对关系数据库的映射规则以及一些对 XML 文档进行验证需要的业务规则等.这些元数据信息使得应用程序可以对 XML 文档作进一步的应用相关的处理.SAF 用 XPath 表达式选取节点集合,用一个符合外部 Schema 的 XML 片断表示应用相关的信息.在 XSDL 中,这个外部 Schema 就是 OWL 的 XML 语法.XSDL 是一个基于 XML 的语言,它的文档结构是基于 SAF 的,如下所示:

```
〈schema-adjunct xmlns:owlx=“http://www.w3.org/2003/05/owl-xml” target=“http://foo.org/myschema.xsd”〉
```

```
〈document〉〈!—全局本体定义:任意 OWL DL 的合法语法--〉
    〈owlx:Ontology owlx:name=“http://foo.org/wine”〉...〈/owlx:Ontology〉
〈/document〉
〈!—映射规则定义:把 XML 中的节点映射到全局本体--〉
〈element context=“/wineMerchant”〉
    〈owlx:Class owlx:name=“WineMerchant”/〉〈!—或:ObjectProperty, DatatypeProperty--〉
〈/element〉...
〈attribute context=“/wineMerchant/wine/name”〉
    〈owlx:DatatypeProperty owlx:name=“name”/〉〈!—或:ObjectProperty--〉
〈/attribute〉...
〈/schema-adjunct〉
```

其中,属性 target 值是要描述的 XML Schema 的 URL,称为目标模式.语义信息在文档、元素节点和属性节点都可描述:document 元素的内容是一个全局本体的声明,可以包含任意 OWL DL 语法允许的成分,这个全局本体也称为 XSDL 中的本体;element 节点和 attribute 节点是对映射规则的定义,其属性 context 的值是一个 XPath 表达式,表示选取的节点集合,其内容是对全局本体中个体、类、对象属性和数据属性的引用.

由此可见,XSDL 通过 SAF 把 XPath 表达式和 OWL DL 的 XML 语法结合起来,充分利用了 XPath 和 OWL DL 丰富的表达能力,通过 XPath 选取具有语义的节点,而语义又可以通过 OWL DL 描述出来.XSDL 是在 Schema 层次上定义的,因此,XSDL 定义可以应用于符合这个 Schema 的所有 XML 文档.但是,为了直观起见,在介绍 XSDL 的语法时所用的例子都是 XML 片断.

2.2 类定义

在 XML 中,一个元素节点往往代表着一个类的实例,这些节点的集合代表着一个类.在例 1 中,wine 节点的集合表示了一个 Wine 类,这可以在 XSDL 中表示为

```
〈element context=“/wineMerchant/wine”〉
    〈URIFunction〉concat(“http://foo.org/wine#”, string(“/wineMerchant/wine[$i]/@id”))〈/URIFunction〉
    〈owlx:Class owlx:name=“Wine”/〉
〈/element〉
```

其中,context 属性值是 XPath 路径表达式,回文档中的 wine 节点集合;URIFunction 节点表示一个构造 URI 的 XPath 函数;参数$*i* 表示 context 返回的节点集合的第 *i* 个节点;string 和 concat 都是 XPath 的内置函数.引入 URI 构造函数的原因是,由于在 OWL 本体中,实例一般都有 URI 标识,URI 构造函数可以利用具有 ID 性质的属性的值,这正好可以解决在 OWL DL 的一个缺陷,即数据属性不能声明为反函数型.通过把 ID 属性的值加入 URI 的构造函数,就可以用 URI 标识实例.如果没有给出 URI 函数,这些节点将会被解释为匿名资源.

类定义的抽象语法是一个三元组:〈CtxPath^^element, urifn, cn^^Class〉.其中,CtxPath 是 context 属性的值,^^element 表示 CtxPath 选取的节点类型是元素节点,urifn 是 URI 构造函数,cn 是本体中类的名字,^^Class 表示 cn 是一个类的名字.

2.3 数据属性定义

在 XML 中,属性节点和一些类型 PCDATA 的元素节点往往表示数据属性.在 XSDL 中,除了需要定义属性的定义域和值域以外,还需要定义 domainContext 的 path 属性,即定义域对应的 XPath 路径表达式;同时也需要定义 rangeContext 的 path 属性,即值域对应的 XPath 路径表达式,它们都必须是相对于 context 的相对路径.这样可以确认定义域中的一个实例与值域中的哪个值是通过这个数据属性关联的.也就是说,XSDL 不仅要定义 XML Schema 对应的本体,而且还定义了具体的实例和值是怎样通过属性关联的.这样,对于一个符合此 Schema 的 XML 文档,根据 XSDL 的定义,就能生成每个数据属性对应的〈实例,值〉二元组.

在例 1 中,元素节点 name 都表示了一个类 Wine 的数据属性,这可以在 XSDL 表示为

〈element context=“/wineMerchant/wine/name”〉

〈domainContext path=“..”/〉 〈rangeContext path=“text()”/〉

〈owlx:DatatypeProperty owlx:name=“wineName”/〉

〈/element〉

其中,属性 wineName 已定义在全局本体中.

数据属性的抽象语法是一个四元组:〈CtxPath^^nodeType,DPath,RPath,dpn^^DatatypeProperty〉. 其中,“nodeType”可以是“element”和“attribute”,DPath 和 RPath 分别是 domainContext 节点和 rangeContext 节点的 path 属性,dpn 是数据属性的名字.

2.4 对象属性表示

在 XML 中,元素节点的嵌套常常表示对象属性.在例 1 中,wineMerchant 节点和 wine 节点的嵌套表示了一个销售(sell)的对象属性.这在 XSDL 中的定义和数据属性的定义是类似的.另外,对象属性可以通过 IDREF 类型的属性节点和元素节点对其他节点的引用来表示.

例 2:关于交叉引用的 XML 片断.

〈wine id=“w1001” name=“Vielles Bottes” color=“black”/〉

〈wineMerchant name=“Bristol Bottlers” wineID=“w1001”/〉

其中,wineID 属性是对元素 wine 的 ID 属性节点的引用.通过这个引用,销售商和酒建立了销售关系,要在 XSDL 中表示这种引用,还需要定义 IDPath 和 IDREFPath,分别表示被引用的节点路径和表示引用的节点路径.为了方便实现,IDPath 属性值应该是绝对路径表达式,而 rangeContext 的 path 属性值应该是相对于 IDPath 的相对路径;IDREFPath 属性值应该是相对于 context 的相对路径.其 XSDL 定义如下:

〈attribute context=“/wineMerchant/@wineID”〉

〈domainContext path=“..”/〉

〈rangeContext path=“..” IDPath=“/wine/@id” IDREFPath=“.”/〉

〈owlx:ObjectProperty owlx:name=“sell”/〉

〈/attribute〉

对象属性定义的抽象语法是一个六元组:〈CtxPath^^nodeType,DPath,RPath,IDPath,IDREFPath,opn^^ObjectProperty〉.其中,opn 是对象属性的名字.如果没有引用,IDPath 和 IDREFPath 也不会出现.

3 XML 的模型论语义

在 XML 有了 XSDL 描述它的语义信息以后,XML 文档不但可以看成数据的载体,而且还表达了数据的语义.下面,我们通过定义 XML 的模型论语义把这些语义信息显式地表示出来.XML 的模型论语义是定义在 XML 的数据模型上的.它是 XML 中所包含的数据信息的抽象表示.我们首先定义 XML 的简单解释,然后定义 XML 的 XSDL-解释,即把 XSDL 理解为对 XML 简单解释的扩展和限制.基于 XML 形式化语义,就可以讨论 XML 中的语义有效性和推理问题了.

3.1 XML的简单解释

XML 文档在经过 XML Schema 的语法验证后,生成一个数据模型,包含了文档中各种数据信息.我们把 XML 的数据模型作为 XML 语义解释的词汇集.由于数据模型中包含了数据类型信息,我们首先给出 XML 数据类型的解释.

定义 **1**(数据类型). 一个数据类型 d 由以下元素来定义:

1) 一个非空的字符串集合,称为 d 的词法空间,记为 $L(d)$;

2) 一个非空的值集合,称为 d 的值空间,记为 $V(d)$;

3) 一个映射 $L2V(d):L(d)\rightarrow V(d)$,即把具有数据类型 d 的字符串映射到对应的值.

另外,定义一个数据类型映射 D,它把数据类型名称映射到数据类型.

定义 **2**(**XML** 词汇集). 一个 XML 的词汇集 V 是这个 XML 文档的数据模型,它包含如下元素:

1) 节点集合 $N=N_e\cup N_a\cup N_t$,其中,N_e 表示元素节点集合,N_a 表示属性节点集合,N_t 表示文本节点集合.在 XML 的语义解释中,忽略其他类型节点的意义.
2) 节点对的集合 $NP=N\times N$,它是定义在节点集合 N 上的一个二元组的集合,包含了由文档序决定的节点序偶的集合.

定义 **3**(简单解释). 一个 XML 文档关于其词汇集 V 的简单解释 I 包含如下元素:

1) R:非空的全体资源的集合,是解释 I 的论域.
2) LV:全体文字值(literal value)的集合.它是 R 的一个子集,包含了平凡文字(plain literal)的集合以及所有数据类型的值空间.
3) O:解释 I 的全体实例的集合.它是 R 的一个子集,且与 LV 不相交.
4) S:$V_{I0}\to O$:把 XML 文档中出现的 URI(记为 V_{I0})都映射为实例,这些 URI 可能是作为属性和文本节点的值.
5) Mc:$N_e\to O\cup LV$:把元素节点映射到实例和文字值,但并不是每个元素节点都有映射.
6) Mc:$N_a\cup N_t\to LV$:把所有的属性节点和文本节点映射为实例或文字值,对任意 $m\in N_a\cup N_t$:
   a) 如果 m 是 OWL 支持的数据类型,则 $Mc(m)\in LV$,$Mc(m)=dm$:*typed-value*(m);
   b) 如果 m 是 OWL 不支持的数据类型,则 $Mc(m)\in V(D(xsd$:*string*$))$,$Mc(m)=dm$:*string-value*(m).
7) Mo:$NP\to R\times R$:把节点对映射为资源的二元组,即对$\langle m,n\rangle\in NP$, $Mo(\langle m,n\rangle)=\langle Mc(m),Mc(n)\rangle$.

XML 文档的简单解释已经初步为节点和节点序偶赋予了一定的意义,比如,节点表示了一个实例或文字值,节点序偶表示了一种实例或文字值之间的关系,但并没有指明这个实例的类型以及关系具体是什么关系.这些进一步的信息实际上在其 Schema 的 XSDL 定义中有描述.下面,我们把一个 XML 文档关于其 Schema 的 XSDL 定义的解释简称为 XML 的 XSDL-解释.

3.2 XML的XSDL-解释

XSDL 定义了 XML 文档作者的本义,因此,对 XSDL 的解释是对 XML 简单解释的扩展.首先,我们定义 XSDL 中的词汇集,然后再定义 XSDL-解释.

定义 **4**(**XSDL** 词汇集). 一个 XSDL 的词汇集包含如下元素:

1) V_0:XSDL 中的全局 OWL DL 本体的词汇集,包含了 V_L,V_C,V_D,V_I,V_{DP},V_{IP},V_{AP} 和 V_O,其各部分的含义可参考 OWL 模型论语义[11];
2) XP:一个 XPath 路径表达式的集合 $XP=AXP\cup RXP$,其中,AXP 表示绝对路径表达式集合,RXP 表示相对路径表达式集合;
3) FN:XSDL 中,类定义中的 URI 构造函数的集合.

定义 **5**(**XSDL**-解释). 一个 XML 文档的 XSDL-解释 I 在 XML 的简单解释基础上:

1) 增加了 Map:$AXP\to P(N)$:把 XPath 的绝对路径表达式映射到节点的集合,并且映射到的节点集合是按照 W3C 的 XPath 规范正确得到的(其中 P 是集合的幂运算符);
2) 增加了 Mrp:$N\times RXP\to P(N)$:把 XPath 的相对于某节点的相对路径表达式映射到节点的集合,并且映射到的节点集合是按照 W3C 的 XPath 规范正确得到的;
3) 增加了 M_{fn}:$N\times FN\to V_{IX}$:把相对于节点 n 的 URI 构造函数映射为一个 URI,其中 V_{IX} 是这些函数返回的 URI 的集合;
4) 扩展了 S:$V_{IX}\cup V_0\to R$:S 还把 XSDL 中的 OWL DL 本体的词汇集 V_0,以及把 URI 构造函数返回的 URI 集合 V_{IX} 映射到资源集合 R,且满足 $S(V_{IX})\subseteq O$,即把通过 URI 构造函数构造的 URI 都映射到论域中的实例;
5) 增加了 XSDL 定义的 OWL 本体的一个解释:$I_0=(R_0,LV,O_0,S,L,EC,ER)$,其中,$O_0=S(V_I)$;$R_0=S(V_0)$;$L,EC,ER$ 分别是 OWL 模型论语义[11]对全局本体中的类型文字、类和属性的解释.

在 XSDL 解释中,一个 XPath 路径表达式被映射到一个节点集,进而这些集合中的节点被简单解释中的 Mc 映射到论域中的实例或文字值.为了简单起见,我们没有具体分析 XPath 表达式的语法和语义.

由 XSDL 中的定义进一步说明了节点的含义以及节点表示的资源之间的关系,它们被解释为 XML 的 XSDL-解释上的语义条件,即:

1) 如果 XSDL 中有类定义三元组⟨CtxPath^^element,urifn,cn^^Class⟩,则 $cn\in V_c$,$Map(CtxPath)\subseteq N_e$,对任意 $n\in Map(CtxPath)$,使得 $M_{fn}(n,urifn)\in V_{IX}$,$S(M_{fn}(n,urifn))=Mc(n)$,$Mc(n)\in EC(cn)$,即节点 n 解释为名为 cn 的类的实例,这个实例的 URI 是 URI 构造函数以节点 n 为参数返回的 URI.
2) 如果 XSDL 中有数据属性定义⟨CtxPath^^nodeType,DPath,RPath,dpn^^DatatypeProperty⟩,则 $dpn\in V_{DP}$,对任意 $n\in Map(CtxPath)$,满足 $|Mrp(n,DPath)|=1$ 或 $|Mrp(n,RPath)|=1$.即定义域和值域中的节点不能同时有多个,因为这会造成无法确认节点之间的关联关系.设数据属性的值域是数据类型 d,则对任意 $m\in Mrp(n,DPath)$,$t\in Mrp(n,RPath)$,使得 $Mc(t)\in V(d)$,$Mc(t)=L2V(d)(dm{:}string\text{-}value(t))$,$Mo(\langle m,t\rangle)=\langle Mc(m),Mc(t)\rangle\in ER(dpn)$),即节点 t 解释为名为 d 的数据类型的值空间里的元素.节点序偶⟨m,t⟩的解释为节点 m 映射到一个具有数据属性 dpn 的实例,且属性值为节点 t 映射到的文字值.
3) 如果 XSDL 中有引用的对象属性表示:⟨CtxPath^^nodeType,DPath,RPath,IDPath,IDREFPath,opn^^ObjectProperty⟩,则 $opn\in V_{IP}$,对任意 $n\in Map(CtxPath)$及任意 $p\in Mrp(n,IDREFPath)$,存在唯一的 $q\in Map(IDPath)$,满足 $Mc(p)=Mc(q)$,从而存在唯一的 q 的祖辈节点 $k\in Mrp(q,RPath)$,对任意 $m\in Mrp(n,DPath)$,使得 $Mo(\langle m,,k\rangle)=\langle Mc(m),Mc(k)\rangle\in ER(opn)$,即节点序偶⟨$m$,,$k$⟩解释为节点 m 映射到一个具有对象属性 opn 的实例,且属性值为节点 k 映射到的实例.

定义 **6**(**XML** 断言集). 一个 XML 文档表示的断言集是指在 XML 的 XSDL 解释 I 的定义中,解释 I 要满足的一些形如 $Mc(n)\in EC(cn)$或 $Mc(t)\in V(D(ddd))$或$\langle Mc(m),Mc(n)\rangle\in ER(pn)$的断言的集合.

这些断言与解释 I 对本体的解释是否一致呢?例如,假设在全局本体中声明了类 *Wine*,如果根据 XSDL 定义,解释 I 要满足 $Mc(n)\in EC(Wine)$,但如果解释 I 对类 *Wine* 的解释 $EC(Wine)$不包含 $Mc(n)$,显然,这个解释是不能同时满足这个断言和全局本体的.下面,我们给出 XML 模型的定义.

定义 **7**(**XML** 模型). 一个 XML 文档的 XSDL-解释 I 是这个 XML 文档的模型,如果:

1) 解释 I 满足 XSDL-解释上的语义条件;
2) 解释 I 是 XSDL 定义中的全局本体的模型.

因此,如果一个 XML 文档的 XSDL-解释是这个文档的模型,则这个文档表示的 XML 断言集和 XSDL 中的全局本体是一致的.值得一提的是,XML 的模型只有在相对于它的 XSDL 定义的前提下才有意义.

3.3 XML文档的语义有效性

有了 XML 模型的概念以后,我们就可以定义 XML 文档的语义有效性了.它是 XML 文档的一个重要特征,说明了 XML 文档表达的语义是否含有矛盾,即是否满足一些 XML Schema 无法表达的语义完整性约束.

定义 **8**(语义有效性). 一个 XML 文档关于其 XSDL 定义是语义有效的,如果这个 XML 文档存在一个模型.

为了检查 XML 文档的语义有效性,我们首先引入 XML 的“对应本体”.我们知道,XSDL 定义中的 OWL DL 本体对应于$\mathcal{SHOIN}(\mathcal{D})$表示的知识库,其公理部分对应于 TBox,事实部分对应于 ABox.而符合此 XSDL 定义的 XML 文档表示的断言集也对应于关于此 TBox 的 Abox.因此,XML 文档和本体的事实部分可以合并,加上原来的公理部分,可构成一个新的本体.

定义 **9**(对应本体). 一个 XML 文档及其 XSDL 定义的对应本体是指 XSDL 定义中包含的 OWL DL 本体在合并上 XML 文档表示的断言集后得到的新本体.有时简称为 XML 的对应本体.

我们将使用语言的扩展和模型的膨胀的定义[12].一个逻辑语言(如一阶语言)L_0是语言 L 的扩展,如果 L 中的每个非逻辑符也在 L_0 中.一个理论 T_0 是理论 T 的扩展,如果 T_0 对应的语言 $L(T_0)$是 $L(T)$的扩展,且 T 中的每个公理也在 T_0 中.假设 I_0 是 L_0 的一个解释,通过忽略一些非逻辑符的解释,可以得到一个解释 I.我们称 I 是 I_0 在 L 上

的限制,I_0 是 I 到 L_0 的膨胀.有如下引理:

引理 **1**. 如果理论 T 是 T_0 的扩展,M 是 T 的模型,则 M 限制在 T_0 的语言上也是 T_0 的模型[12].

在我们的问题中,XML 加上 XSDL 可看成是一个语言,一个 XML 文件及其 XSDL 定义则是这个语言的一个理论.显然,这个语言是本体语言 OWL DL 的扩展,其理论也是 OWL DL 语言的理论的扩展.

然后,我们引入一个关于 XML 的模型和本体模型的关系的引理.

引理 **2**. XML 的模型在 OWL DL 语言上的限制是其对应本体的模型;XML 对应本体的模型可以膨胀到此 XML 的模型.

最后,我们可以得到判断 XML 语义有效性的定理.

定理 **1**. 一个 XML 文档是语义有效的,当且仅当对应的本体是可满足的.

3.4 XML文档的语义蕴涵

在 XML 具有模型论语义之后,我们可以定义 XML 中的推理,其中最重要的概念是蕴涵.但由于 XML 的节点集合在语言的词汇集中,如果 XML 文档 D_1 和 D_2 具有不同的节点,则 D_1 的模型不可能也是 D_2 的模型,因为在 D_1 中没有对 D_2 中不同节点的解释.因此,当我们定义 XML 文档之间的蕴涵时,要对模型进行膨胀,使模型也会对 D_2 中不同的节点进行解释.

定义 **10**(语义蕴涵). 假设 XML 文档 D_1 和 D_2 符合同一 Schema,且具有相同的 XSDL 定义,称 D_1 蕴涵 D_2,如果任意 D_1 的模型都存在一个膨胀是 D_2 的模型.记为 $D_1 \models D_2$.

为了把 XML 的蕴涵问题规约到本体的蕴涵问题,我们有如下定理:

定理 **2**. 记 XML 文档 D_1 和 D_2 对应的本体分别是 O_1 和 O_2,则 D_1 蕴涵 D_2,当且仅当本体 O_1 蕴涵本体 O_2,即 $D_1 \models D_2$ iff $O_1 \models O_2$.

定理 **3**. OWL DL 本体的蕴涵推理问题可以转化为描述逻辑语言$\mathcal{SHOIN}(\mathbf{D})$中的不可满足性问题[10].

推论 **1**. XML 文档的蕴涵问题可以转化为描述逻辑语言$\mathcal{SHOIN}(\mathbf{D})$中的不可满足性问题.

要注意的是,描述逻辑语言$\mathcal{SHOIN}(\mathbf{D})$中的不可满足性问题的时间复杂度是 NExpTime,目前并没有完全实现其推理的推理机.但如果本体语言限制为 OWL Lite,则 XML 文档的蕴涵问题可以转化为描述逻辑语言$\mathcal{SHIF}(\mathbf{D})$中的不可满足性问题,其复杂度是 ExpTime.而且,推理机(如 Racer)能够提供高效的推理服务.

4 相关工作

关于 XML 的“语义”具有不同的理解.Psaila 等人[13]最早于 1999 年注意到由 XML 标记的文档和由 BNF 语法生成的字符串之间的类似性,提出了一种把 DTD 的元素描述转换为形式化 EBNF 规则的方法.在 SGML 领域,BECHAMEL 项目[14]也提出了 XML 语义的概念.该项目试图用知识表示的工具表示 XML 标签表示的语义信息,其原型实现系统是基于 Prolog 的[15].但是,这些工作都没有给出 XML 的形式化语义.

最近,Patel-Schneider 和 Simeon[7]提出了阴阳 Web 的概念,统一了 XML 和 RDF 的数据模型和语义.然而,由于 XML 作者能以几乎任意的方式表达语义,像阴阳 Web 那样直接给出的 XML 语义解释很难与作者的本意符合.另外,直接给出的解释局限于 XML 的文档结构,从而引入了过多的关系,但又无法表达跨层的节点关系以及有条件的表示.我们引入 XSDL 解决了这个问题,因为 XSDL 可以让作者明确地说明 XML 中包含的语义信息.我们对 XML 的解释更为清晰,且能够捕捉到作者的本意.

XSDL 和 MDL 都采用了模式附件框架和用概念模型定义 XML 语义.但是,MDL 的语法是专有的,并且它的概念模型是基于 UML 的.相反地,XSDL 的语法大部分都是基于标准的 XPath 和 OWL 语法.另外,XSDL 采用 OWL DL 作为模型语言,从而具有形式化的语义,有助于弥补 XML 和语义 Web 之间的差距.

XSDL 通过 XML 到本体的映射来定义 XML 语义,这也有一些相关工作.例如,Amann 等人[16]提出了一种映射规则语言,把 XML 的元素映射到本体的概念;Erdmann 等人[17]则提出了一种从本体生成 DTD 的算法,这样,符合这个 DTD 的 XML 文档中的标签就能和本体关联起来.

5 结束语

在本文中,为了解决 XML 缺乏形式化语义的问题,我们提出了 XML 的语义定义语言 XSDL,并给出了一种 XML 的模型论语义.XSDL 是一种简单、易懂的语言,其语法大部分是基于标准的 XPath 和 OWL 的 XML 语法,以及模式附件框架 SAF.更为重要的是,我们提出了 XML 的一种模型论语义,它分简单解释和 XSDL 解释两步,能够表达 XML 文档作者的本意.这个语义和 OWL DL 支持的 RDF 部分的语义是兼容的.因此,XML 可以看成是 RDF 的一个子语言.此外,我们还提出了 XML 的语义有效性概念以及 XML 文档的推理问题,并把它们规约到描述逻辑的推理问题,从而可以利用描述逻辑的推理机实现对 XML 的推理.

与阴阳 Web 一样,在 XML 具有显式的、机器可理解的语义信息以后,用户和程序就可以在语义的层次上和 XML 交互,为 XML 带来了很多新的应用场景.例如,对 XML 的语义查询,即在概念层次上对 XML 进行查询,而不关心 XML 的语法和结构.另一个应用是 XML 的数据集成,可以通过 XSDL 定义把不同的数据源都映射到全局的本体上.此外,更为重要的是,通过 XSDL 的定义,XML 可以很容易地在保持语义的条件下转换为 RDF,这为解决语义 Web 的六大挑战之首(即具有语义内容的获取)跨进了一步,对当前 Web 向语义 Web 的过渡具有重要的意义.我们将在下一步的工作中研究这些问题.

References:

[1] Cover R. XML and semantic transparency. 1998. http://www.oasisopen.org/cover/xmlAndSemantics.html

[2] Uschold M. Where are the semantics in the semantic Web. AI Magazine, 2003,24(3):25−36.

[3] Berners-Lee T, Handler J, Lassila O. The semantic Web. Scientific American, 2001,184(5):34−43.

[4] Klyne G, Carroll JJ. Resource description framework (RDF): Concepts and abstract syntax. W3C Recommendation, 10, 2004. http://www.w3.org/TR/rdf-concepts/

[5] Thompson HS, Swick R. The cambridge communiqué. W3C NOTE 7, 1999. http://www.w3.org/TR/schema-arch

[6] Buswell S, Brickley D, Matthews B. SWAD-Europe deliverable 5.1: Schema technology survey. 2003.

[7] Patel-Schneider PF, Simeon J. The yin/yang Web: A unified model for XML syntax and RDF semantics. IEEE Trans. on Knowledge and Data Engineering, 2003,15(4):797−812.

[8] Worden R. MDL: A meaning definition language, version 2.06. 2002. http://www.charteris.com/publications/whitepapers/

[9] Vorthmann S, Buck L. Schema adjunct framework draft specification. 2000. http://www.tibco.com/software/standards_support/xmlresources/spec.html

[10] Horrocks I, Patel-Schneider PF. Reducing OWL entailment to description logic satisfiability. Journal of Web Semantics, 2004,1(4):345−357.

[11] Patel-Schneider PF, Hayes P, Horrocks I. OWL Web ontology language semantics and abstract syntax. W3C Recommendation, 10, 2004. http://www.w3.org/2001/sw/Europe/reports/xml_schema_tools_techniques_report

[12] Shoenfield JR. Mathematical logic. Addison-Wesley Publishing Company, 1967.

[13] Psaila G, Crespi-Reghizzi S. Adding semantics to XML. In: Parigot D, Mernik M, eds Proc. of the 2nd Workshop on Attribute Grammars and their Applications (WAGA'99). Amsterdam, 1999. 113−132.

[14] Sperberg-McQueen CM, Huitfeldt C, Renear A. Meaning and interpretation of markup. Markup Languages: Theory & Practice, 2000,2(3):215−234.

[15] Dubin D, Sperberg-McQueen CM, Renear A, Huitfeldt C. A logic programming environment for document semantics and inference. Literary and Linguistic Computing, 2003,18(2):225−233.

[16] Amann B, Fundulaki I, Scholl M, Beeri C, Vercoustre AM. Mapping XML fragments to community Web ontologies. In: Giansalvatore Mecca, Jérôme Siméon.eds.. Proc. of the 4th Int'l Workshop on the Web and Databases (WebDB 2001). 2001.

[17] Erdmann M, Studer R. How to structure and access XML documents with ontologies. Data and Knowledge Engineering, 2001, 36(3):317−335.

刘升平(1977－),男,江西分宜人,博士生,主要研究领域为语义 Web.

梅婧(1980－),女,博士生,主要研究领域为语义 Web.

林作铨(1963－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机科学,人工智能.

岳安步(1980－),男,博士生,主要研究领域为常识推理.