文章编号:1001 - 9081(2007)07 - 1715 - 02

一种 DTD 一致性的判定方法

邱长春[1,2], 薛超英[2], 胡罗凯[1]
(1. 湖北教育学院 计算机科学与工程系,武汉 430205; 2. 武汉大学 计算机学院,武汉 430072)
(qccxl9@yahoo.com.cn)

摘 要:DTD 作为一种 XML 文档结构的模式语言得到了广泛的使用,它描述了相似的 XML 文档的结构。DTD 的一致性是指对于一个给定的 DTD,判断是否存在至少有一个 XML 文档满足 DTD。在引入 DTD 一致性的形式化定义的基础上,分析了引起 DTD 不一致性的各种因素,提出了 DTD 一致性的判定方法。

关键词:XML;DTD;一致性
中图分类号: TP311 **文献标志码**:A

Approach to judge consistency of DTDs

QIU Chang-chun[1,2], XUE Chao-ying[2], HU Luo-kai[1]
(1. *Department of Computing Science and Engineering, Hubei Institute of Education, Wuhan Hubei* 430205, *China*;
2. *Computer School, Wuhan University, Wuhan Hubei* 430072, *China*)

Abstract: DTD has been widely used as the schema language for XML documents. A DTD describes the structure of a collection of similar XML documents. The consistency of XML DTDs concerns the question that given a DTD D, is there any finite XML document that conforms to D. The notion of the consistency of DTDs was formalized, and a judging method of the consistency for DTDs was given.

Key words: XML; DTD; consistency

DTD 定义了 XML 文档的语法和整体结构,具有简洁的文法规则、参数实体和拥有众多可用工具等优点。一个语法完全正确的 DTD 可能是不一致的,即不存在任何有效的 XML 文档遵循这个 DTD,这样的 DTD 没有任何实际意义,应当尽可能避免。在实际应用中,小型的 DTD 可以根据常识进行检查以保证一致性,但是对于大型的或是其他的数据模型(例如 ER 模型和面向对象模型)转换的 DTD 很难直接判断是否存在与之对应的 XML 文档,因此需要一种自动检测的判定机制。为此,本文在给出 DTD 一致性的概念的基础上,提出一个判定 DTD 一致性的方法。

1 基本概念

不失一般性,我们把属性和元素都作为元素来处理。字符串都用"string"表示。为此,DTD 的形式化定义如下:

定义 1 DTD 表达式。一个 DTD 表达式递归定义为如下的巴科斯 - 诺尔范式:

$e ::= string \mid n \mid e^+ \mid e^* \mid e? \mid (e_1, e_2, \cdots, e_n) \mid (e_1 \mid e_2 \mid \cdots \mid e_n)$

这里 n 指的是 XML 元素名,e 指的是 DTD 表达式,"::="表示"定义为","|"表示"或"。DTD 表达式是由 XML 元素名和 DTD 表达式(递归定义)使用下面的操作符构成:

1)元组操作符。$(e_1, e_2, \cdots, e_n)$ 表示一个 DTD 子表达式元组。特别地,(e) 称为单字符元组,元组操作符为",",。

2)星号操作符"*"。e^* 表示零个或多个 DTD 子表达式系列。

3)加号操作符"+"。e^+ 表示 1 个或多个 DTD 子表达式系列。

4)可选操作符"?"。$e?$ 表示 0 个或 1 个 DTD 子表达式系列。

5)或(or)操作符"|"。$(e_1 \mid e_2 \mid \cdots \mid e_n)$ 表示在多个 DTD 表达式中选取一个 DTD 表达式。

定义 2 DTD 图。在 DTD 图中,节点为元素或属性名,用边表示父元素和子元素之间的关系。由于 DTD 语法中父元素和子元素之间有"?"、"*"、"+"、"-"和"|"五种约束关系,因此在 DTD 图中有对应的五种边,我们分别标记为"?-edge"、"*-edge"、"+-edge"、"-edge"和"|-edge"。

对于图 1 的 DTD,它们对应的 DTD 图如图 2 所示。

DTD1:
```
<!DOCTYPE a[
 <!ELEMENT a(b)>
<!ELEMENT b(a)>
]>
```

DTD2:
```
<!DOCTYPE a[
<!ELEMENT a(#PCDATA)>
<!ELEMENT b(c)>
<!ELEMENT c(b)>
]>
```

DTD3:
```
<!DOCTYPE a[
<!ELEMENT a(b)>
<!ELEMENT b(c?)>
<!ELEMENT c(a*>
]>
```

DTD4:
```
<!DOCTYPE a[
<!ELEMENT a(b | d)>
<!ELEMENT b(c)>
<!ELEMENT c(b)>
<!ELEMENT d(#PCDATA)>
]>
```

图 1 DTD

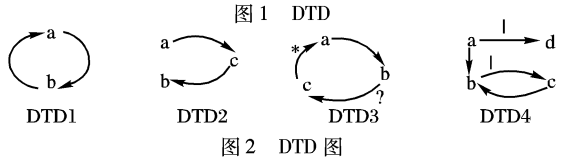

图 2 DTD 图

收稿日期:2007 - 01 - 24;**修回日期**:2007 - 03 - 09。

作者简介:邱长春(1966 -),男,湖北麻城人,讲师,硕士,主要研究方向:XML、数据库、数据挖掘; 薛超英(1958 -),男,浙江舟山人,教授,主要研究方向:数据库、Web 信息系统; 胡罗凯(1981 -),男,湖北武汉人,硕士,主要研究方向:数据库、Web 信息系统。

定义 3 强约束和弱约束。DTD 表达式中,如果元素之间的操作符是“+”和“-”,那么元素之间的约束就是强约束,其他元素之间的约束就是弱约束。在对应的 DTD 图中,就有与之对应的强约束边和弱约束边。根据操作符是“+”和“-”的语义,强约束要求父元素至少有一个子元素,弱约束则不尽然。

定义 4 有效的 XML 文档。遵守 XML 语法规则且遵守相应的 DTD 文档规范的 XML 文档。

尽管有效的 XML 文档同时遵守 XML 文档规范和 DTD,但是语法完全正确的 DTD 并不一定有与之对应的 XML 文档。如图 1 的 DTD1,从对应的 DTD 图可以看出,元素 a 和 b 相互包含,尽管 DTD 的定义符合 DTD 的语法规范,不可能与之对应的 XML 文档。

定义 5 DTD 的一致性。一个 DTD 是一致的,当且仅当至少存在一个有效的 XML 文档遵循该 DTD;如果不存在任何有效的 XML 文档遵循该 DTD,则该 DTD 是不一致的。

2 DTD 一致性的判定条件

可以肯定的是,如果 DTD 对应的 DTD 图没有环,完全可以设计一个或几个与之对应的 XML 文档。为此我们给出定理。

定理 1 如果 DTD 对应的 DTD 图中没有环形结构,则该 DTD 是一致的。

根据 DTD 一致性的定义,定理 1 显然成立。根据定理 1,我们给出如下推论:如果 DTD 不是一致的,则 DTD 中必然存在环形结构。用反证法很容易证明此推论的正确性。

DTD 的不一致性是由 DTD 对应的 DTD 图中的环形结构引起的,但是并不是说,DTD 图中所有的环形结构都能引起 DTD 的不一致性。例如图 1 中的 DTD3 和 DTD4 对应的 DTD 图是环形结构,但是它们分别有这样的 XML 文档与之对应:<a><c>Hello!</c>和<a><d>Hello, World!</d><a>,所以是一致的。但是,DTD3 和 DTD4 的环形结构是完全不相同的。DTD3 的环形结构由弱约束边构成,DTD4 的环形结构由强约束边构成。根据弱约束的语义,这样的环形结构不一定存在;只有强约束才构成真正的环形结构,但是 DTD4 尽管存在强约束构成的环形结构又是一致的,原因是由于 DTD4 中存在操作符“|”,它表示可选操作,因此不一定引起环形结构。由于“|”操作符使得 DTD 结构复杂化,我们把 DTD 分为有 DTD 约束和无 DTD 约束两种情况讨论。

2.1 无“|”约束 DTD 的判定条件

在无“|”约束的 DTD 中,只有强约束操作符才构成真正的环形结构,根据一致性的定义和前面的分析,环形结构使得元素递归引用,使得不能定义相应的 XML 文档,所以是不一致的。为此我们给出无“|”约束 DTD 不一致的判定条件。

定理 2 无“|-free”约束的 DTD 具有一致性的充分必要条件是当且仅当 DTD 对应的 DTD 图中不可能产生强约束循环。

必要性 我们用反证法证明。假定一个 DTD D 是一致的,且从根节点 r 有一个强约束循环:$e_0 \to e_1 \to \cdots \to e_n \to e_0$,那么,满足这样的 DTD 的任何 XML 文档所有的元素必在这个循环里。由于 D 是一致的,那么存在一个 XML 文档 x 满足 DTD D。x 必然包含 DTD 的所有元素:$e_0, e_1, \cdots, e_n$。x 元素的有限性说明在 XML 文档的层次结构里存在一个“最里面”的元素。由于从根节点 r 有一个强约束循环:$e_0 \to e_1 \to \cdots \to e_n \to e_0$,也就是循环中要么是强约束边“-edge”,要么是强约束边“+-edge”。所以 e_i 至少包含一个 e_{i+1},这与 e_i 是“最里面”的元素相矛盾。因此,不存在这样的 XML 文档 x 满足 DTD D,所以 DTD D 是不一致的。

充分性 假定 DTD 图中不可能产生任何强约束循环,下面我们通过构造一个满足这样 DTD 的 XML 文档来证明 D 是一致的。首先我们通过这样的转换规则来将 D 转换为 D',用 empty 表示空元素,即:$e? \equiv e \mid \text{empty}$,对于给予 DTD 的表达式 $e, e_1, e_2, \cdots, e_n$ 有这样的转换:

1) $e^+ \Rightarrow e$;

2) $e^* \Rightarrow \text{empty}$;

3) $e? \Rightarrow \text{empty}$;

4) $(e_1 \mid e_2 \mid \cdots \mid e_n) \Rightarrow e_1$。

显然,经过这样的转换后,D' 的 DTD 图只包含“-edge”边,此外,D' 的 DTD 图中的强连通子图 g' 必定不是闭合的,因为在 D 的 DTD 图中不可能产生强约束的闭合循环。由于 g' 是非闭合的,那么可以创建一个满足 g' 的 XML 文档 x,因而 x 满足 D'。又满足 D' 的任何文档必定满足 D,所以 x 相对于 D 来说是有效的 XML 文档,因此 D 是一致的。

2.2 “|”约束 DTD 的判定条件

为了处理“|”操作符,我们把具有“|”约束的 DTD D 分解成多个(有限)无“|”约束的 DTD 的集合 $D_1, D_2, \cdots, D_n$。这样根据 DTD 一致性的定义,如果 $D_1, D_2, \cdots, D_n$ 中有一个 DTD 是一致的,那么 D 就是一致的。

定义 6 DTD 表达式的分解。DTD 表达式 e 的分解是无“|”约束的 DTD 分解表达式 $split(e)$ 的集合。$split(e)$ 递归定义如下:

1) $split(string) = \{string\}$;

2) $split(n) = \{n\}$;

3) $split(e^+) = \{g^+ \mid g \in split(e)\}$;

4) $split(e^*) = \{g^* \mid g \in split(e)\}$;

5) $split(e?) = \{g? \mid g \in split(e)\}$;

6) $split((e_1, e_2, \cdots, e_n)) = \{(g_1, g_2, \cdots, g_n) \mid g_i \in split(e_i), i = 1, 2, \cdots, n\}$;

7) $split((e_1 \mid e_2 \mid \cdots \mid e_n)) = split(e_1) \cup split(e_2) \cup \cdots \cup split(e_n)$

我们分解的定义扩展到 DTD 元素及 DTD。

定义 7 DTD 分解。一个 XML DTD $D = \{d_1, d_2, \cdots, d_k\}$ 的分解定义为:$split(D) = \{\{d'_1, d'_2, \cdots, d'_k\} \mid d'_i \in split(d_i), i = 1, 2, \cdots, k\}$。

例如图 1 中的 DTD4 的 DTD 表示为 $D = \{a = b \mid d, b = c, c = b, d = string\}$,那么 D 的分解为:$split(D) = \{\{a = b, b = c, c = b, d = string\}, \{a = d, b = c, c = b, d = string\}\}$。注意到在 $split(D)$ 中的第一个分解是不一致的,因为从根节点 a 导出了一个闭合循环;但是第二个分解是一致的。

基于上面的推理,我们得出对于任意一个 DTD 一致的充分必要条件。

定理 3 DTD 是一致的充分必要条件是 DTD 的分解 $split(D)$ 中至少有一个分解表达式是一致的。

(下转第 1724 页)

引擎由于带有知识库,能够将效果的设置反映到知识库中去。

OntoComposer 平台将领域本体的解析、语义 Web Service 的可视化组装和 Web Service 的 OWL-S 描述的执行有机地结合起来。在领域知识层进行可视化组装是本文所介绍的组装方式的另一个特点。组装人员并不需要关于 Web Service 技术的知识,也不需要具有编程能力。而目前较流行的另一组装工具,OWL-S Editor[15],由于在 Web Service 的实现层进行组装,因此需要有 Web Service 和 OWL-S 的知识,并且组装过程较为复杂。如果单独用 OWL-S API 进行组装,则需要进行编程。

致谢 感谢我的导师崔光佐教授的悉心指导和刘杨师兄为 OntoCompser 所做的出色工作。

参考文献:

[1] W3C. Web Service Activity [EB/OL]. [2006 - 11 - 01]. http://www.w3.org/2002/ws/#drafts.

[2] MILANOVIC N, MALEK M. Current solutions for web service composition [J]. IEEE Internet Computing, 2004, 8(6):51 - 59.

[3] W3C. Semantic web activity [EB/OL]. [2006 - 12 - 31]. http://www.w3.org/2001/sw/.

[4] MARTIN D, BURSTEIN M, HOBBS J, *et al*. OWL-S: semantic markup for web services [EB/OL]. [2004 - 11 - 30]. http://www.w3.org/Submission/OWL-S/.

[5] 刘杨. 基于本体的混合式语义 Web 服务组装机制及原型实现[D]. 北京:北京大学, 2006.

[6] MCGUINNESS D L, VAN HARMELEN F. OWL web ontology language overview[EB/OL]. [2004 - 02 - 28]. http://www.w3.org/TR/owl-features/.

[7] 宋炜,张铭. 语义网简明教程[M]. 北京: 高等教育出版社, 2004.

[8] 梅婧,刘升平,林作铨. 语义 Web 语言的逻辑分析[C]// 第二届全国智能信息网络学术会议论文集. 谭铁牛. 北京:科学出版社,2004.

[9] AKKIRAJU R, FARRELL J, MILLER J, *et al*. Web Service Semantics - WSDL-S[EB/OL]. [2005 - 11 - 01]. http://www.w3.org/Submission/WSDL-S/.

[10] HORROCKS I, PATEL-SCHNEIDER P F, BOLEY H, *et al*. SWRL: A semantic web rule language combining OWL and RuleML[EB/OL]. [2004 - 03]. http://www.w3.org/Submission/SWRL/.

[11] KLYNE G, CARROLL J J. Resource description framework (RDF): Concepts and abstract syntax[EB/OL]. [2004 - 02 - 01]. http://www.w3.org/TR/rdf-concepts/#section-data-model.

[12] Graphviz-graphvisualizationsoftware[EB/OL]. [2006 - 12 - 01]. http://graphviz.org/.

[13] Maryland Information and Network Dynamics Lab. Semantic web agents project. OWL-S API[EB/OL]. [2006 - 12 - 01]. http://www.mindswap.org/2004/owl-s/api/.

[14] RESIN C[EB/OL]. [2006 - 12 - 01]. http://www.caucho.com/.

[15] The OWL-S Editor [EB/OL]. [2006 - 12 - 01]. http://owlseditor.semwebcentral.org/.

[16] World Wide Web Consortium [EB/OL]. [2006 - 12 - 01]. http://www.w3.org/.

[17] TRAVERSO P, PISTORE M. Automated composition of semantic web services into executable processes [C]// Proceedings of ISWC'04. Hiroshima, Japan: [s.n.], 2004:380 - 394.

[18] SIRIN E, HENDLER J, PARSIA B. Semi-automatic composition of Web services using semantic descriptions[C]// Proceedings of Web Services: Modeling, Architecture and Infrastructure workshop in conjunction with ICEIS 2003. 2002:17 - 24.

[19] RAO JH, SU XM. A survey of automated web service composition methods [C]// Proceedings of Semantic Web Services and Web Process Composition workshop. CA, USA: Springer, 2004.

[20] (美) RUSSELL S, NORVIG P. 人工智能——一种现代方法[M]. 2 版. 姜哲,张敏,杨露,等译. 北京:人民邮电出版社, 2004.

[21] SRIVASTAVA B, KOEHLER J. Web service composition - current solutions and open problems[C]// ICAPS 2003 Workshop on Planning for Web Services. Trento, Italy: [s.n.], 2003.

[22] COSTA LD, PIRES P, MATTOSO M. Automated composition of web services with contingency plans[C]// IEEE International Conference on Web Services. USA: IEEE Computer Society, 2004.

[23] Description Logic[EB/OL]. [2006 - 12 - 01]. http://dl.kr.org/.

[24] Axis[EB/OL]. [2006 - 12 - 01]. http://ws.apache.org/axis/.

(上接第 1716 页)

有了上面的定理,判断 DTD 是否具有一致性的算法就非常直接简单了:首先对 DTD D 进行遍历,判定 DTD 是否存在“|”操作符,若存在,则对 DTD D 进行分解,使得分解后的 DTD 不受“|”的约束。接着对分解的 DTD 创建 DTD 图,然后对每一个DTD图从根节点开始沿着强约束边做深度优先遍历,每个第一次访问的节点做“visited”的标记,当有节点访问两次时,就表明出项了闭合回路,那么 D 就是不一致的,否则,D 就具有一致性。

3 结语

鉴于在 DTD 的定义中存在不合理的结构的可能,本文提出了 DTD 一致性的概念,对能够导致 DTD 不一致的各种因素进行了分析,给出了 DTD 一致性的判定条件,这些研究能够在快速判断 DTD 结构设计是否合理等方面提供了有效的帮助。由于 XML Schema 作为一种新的结构模式,具有 DTD 不可比拟的优越性,所以今后的工作是 XML Schema 一致性和 XML 键及 XML 结构模式相互作用的完整性约束。

参考文献:

[1] FAN W, LIBKIN L. On XML integrity constraints in the presence of DTDs[C]// Proceedings of ACM Symposium on Principles of Database Systems. New York: ACM Press, 2001: 114 - 125.

[2] ARENAS M, FAN W, LIBKIN L. On verifying consistency of XML specification[C]// Proceedings of 21st ACM Symposium on Principles of Database Systems. New York: ACM Press, 2002: 259 - 270.

[3] ARENAS M, FAN W, LIBKIN L. What's hard about XML schema constraints?[C]// Proceedings of 13th International Conference on Database and Expert Systems Applications. Berlin: Springer-Verlag, 2002: 269 - 278.

[4] BUNEMAN P, DEVIDSON S, FAN W, *et al*. Reasoning about keys for XML[C]// Database Programming Languages, the 8th International Workshop. Frascati: Springer - Verlag, 2001:133 - 148.