网页树形结构自动生成研究①

Automatic Generation Method for Making Tree-Like Structure in Web Development

张华兴　孙　毅　单继宏　(浙江工业大学 机械制造及自动化教育部重点实验室 浙江 杭州 310014)

摘　要： 网页树形结构是网页语言的一种特定描述。根据网页语言这种规整特性，可结合文法理论形式化建模。通过扩展终结符、规则概念，提出非终结符函数、终结符变量函数的构造方法，给出了自动生成特定类型的语言的一种生成模型。最后，给出模型在仪表装配树形中的实例应用。

关键词： 树形结构 文法 自动生成 生成模型 动态树

树型结构具有界面结构性强、层次好、使用方便等特点，是一类应用非常广泛的数据结构。在互联网页上有两种树型结构[1]，即适合小型的静态结构，实现技术简单；管理人员、维护人员可方便增加、删除树节点的动态结构，技术复杂。动态结构的动态更新源于数据库节点信息。

网页上实现动态树形结构有三种方式：直接利用控件，现实方法未知[2]；在显示层调用脚本函数，解读数据库节点信息[3]；节点信息多级编码，程序按照编码输出不同深度节点，为远端用户展现树形目录[4]。

无论采用何种实现方式，在显示树形结构的网页文件中代码的组成要素基本相同。有必要在剖析树型基本结构、伸展原理基础上，归纳目标代码的语法规则，引入文法理论，提出一种构造规整性语言的方法，并给出生成模型。

1　网页树型结构的规整性

网页树型结构的伸展原理是在触发隐藏或显示的控制对象上添加事件，事件关联的脚本函数响应并修改被控制对象(即被隐藏或显示对象)的布局显示、URL等属性值，达到页面丰富的展现效果。

HTML 语言和脚本语言编写的网页树型结构，组成元素可用 div 标签或 table 标签，table 标签控制树形节点间的宽度和间距更简洁，例如图 1 所示：

S 代表一个简化的基本树形结构对象，M 为复杂对象，N 是没有下级结构的普通对象，正常显示即可。在树形结构中，M 对象或者 N 对象至少要出现一次。

```
S: <table>
   M:  O:  <tr>
           T:  <td width=16 height=16>
               J:  <img style='cursor:hand' src=".\\pic\\dotJia.gif"
                   id='image_1_2' onClick="javascript:ShowSub('1_2')">
               </td>
           U:  <td> M27 卡套组件</td>
           </tr>
       P:  <tr id='fold_1_2'style='display:none'>
           F:  <td></td>
           I:  <td>…</td>
           </tr>
   N:  <tr><td>表盖</td></tr>
   </table>
```

图 1　基本树型结构代码示例

由控制对象 O 和被控制对象 P 两部分组成对象 M。控制对象 O 又由触发对象 T 和显示内容 U 组成。T 显示的内容为一张图片 J，作用提示下级结构是否已经展开；点击这个图片可触发伸缩事件。U 表示控制对象 O 需要在网页上显示的内容。

P 表示被控制对象，又可称为下级结构，由两部分组成：F 表示父子层级的缩进距离；I 是个递归表达，相当于 S 对象的下一级树形结构。

利用事件 onClick 关联脚本函数 ShowSub 判断并改变相应对象的属性值，在 J 对象上鼠标点击则触发事件。“M27 卡套组件”、“表盖”是网页为浏览器解释执行后呈现内容，是事物的数据信息，随网页表达主题而变化。

目标代码组成元素基本情况就是这样，至于上下级间距 F 用空格表示或字符“-”表示、有无伸缩图片 J 显性提醒、触发事件是否直接 U 对象上面定义，对

① 基金项目:浙江省自然科学基金(602088)
收稿时间:2008-11-19

© 中国科学院软件研究所 http://www.c-s-a.org.cn

树形整体的对象层次、出现的次序无影响。实际应用的树形结构是在此基础上扩展，通过 M、N 或 I 的反复来呈现网页企图表达的事物信息。由此，仅需三、四种超级文本标签(含数据信息不同)的排列组合即可构成各类树形图。

目标代码的语言，不仅体现了单一性、无歧义性和明确性，而且凸显了层次性、次序性、递归性。凡具备这些特征则称为满足规整性，符合规整性的语言即为规整语言，可用文法形式化描述，并可构造模型自动生成符合语法规则的目标语言。

2 形式语言和生成模型

在形式语言理论中，形式语言是一个字母表上的某些有限长字符串的集合，可以包含无限多个字符串。当代著名逻辑学家鲍亨斯基认为："形式化方法是这样一种方法，它完全撇开符号本身的意义，而根据某些只涉及符号书面形态的转换规则来进行符号操作"。因此，语言生成模型必须具备描述一类语言的能力，同时还能自动产生实例化的语言，即具有产生个性语言的能力。

2.1 文法基本定义

与研究的特定目标语言相对应的文法，称为文法模型(Grammar Model, GMM)，相关定义和规则的形式化描述如下[5,6]:

定义 1. 终结符是字母表中的符号组成的有穷序列字符串，而且是语言的句子中出现的字符。非终结符，是一个语法变量，表示一个语法范畴。

定义 2. 规则，也称重写规则、产生式或生成式，是形如 α -->β 或 α ::= β 的(α，β)有序对，其中 α 称为规则的左部，β 称作规则的右部。

定义 3. 文法 G 定义为四元组(V_N,V_T,P,S)：其中 VN 为非终结符集；V_T 为终结符集；P 为规则(α --> β)的集合，$\alpha \in (V_N \cup V_T)^*$且至少包括一个非终结符，$\beta \in (V_N \cup V_T)^*$；$V_N$,$V_T$ 和 P 是非空有穷集。S 称作识别符或开始符，它是一个非终结符，至少要在一条规则中作为左部出现。V_N和 V_T不含公共的元素，即 $V_N \cap V_T = \Phi$。通常用 V 表示 $V_N \cup V_T$，V 称为文法 G 的词汇表。

定义 4. 设文法 G=(V_N,V_T,P,S)，则称 L(G)=(ω | $\omega \in VT^*$ 且 S (ω}为文法 G 产生的语言。($\omega \in L(G)$，ω 称为 G 产生的一个句子。

定义 5. 设 G=(V_N,V_T,P,S)，若 P 中的每一个产生式 α -->β 满足：α 是一个非终结符，$\beta \in (V_N \cup V_T)^*$，则此文法称为 2 型的或上下文无关的。

2.2 生成模型建模

生成模型(Generation Model, GNM)是以文法为基础实现特定语言处理目标的指令集，体现为动态的操作过程，目的是产生符合语法规范的目的语言。根据程序活动过程中活动实体的重要性和功能特性，拓展文法定义，进一步描述生成模型。

定义 6. GNM 中终结符集的元素是不确定的，随着目标语言的事物信息而变化，理论上可无穷。在其上，凡预先知晓的、给定的、固定不变的元素，称为终结符常量(Terminal Constant, TC)；对不能预知、给定的、变化的称为终结符变量(Terminal Variable, TV)，但是终结符变量的个数是有限的、确定的。

定义 7. 终结符变量函数(TV Function, TVF),执行一个字符串的组合，反映目标语言组成的多样性的特点，与语法规则无关，且与终结符变量一一对应。

定义 8. 非终结符函数(Nonterminal Function, NTF)，执行一个语法规则，不仅反映目标语言的连接组合规则，而且涉及事物数据信息。即，它需满足跟它同构的文法的非终结符元素和规则元素的对应关系同时，还需反映事物数据信息的内部约束关系。

形如 α -->β 的文法规则转化为 NTF 的调用，故 GNM 函数是普通意义函数的子集，有以下特性：

- 设函数返回值为 x，则有：x 为字符串类型。
- 设有两个函数 NTFi, NTFj 返回值 x_i、x_j，且 NTFi 直接调用 NTFj，则$|x_j| \leqslant |x_i|$。

函数体现字符串的连接关系。

定义 9. 生成模型 GNM 定义为(V_{TC},V_{TVF}, P_{NTF},S)四元组，其中：V_{TC} 为 TC 集；V_{TVF} 为 TVF 集；P_{NTF} 为 NTF 集；V_{TC},V_{TVF}和 P_{NTF}是非空有穷集；S$\in$ P_{NTF},是语言生成起点。模型在程序语言级别对操作过程进行了形式化描述。

定义 10. 称 L(G)=(ω | $\omega \in (V_{TC} \cup V_{TVF})^*$ 且 S $\Rightarrow^* \omega$}为生成模型 GNM 产生的目的语言。$\forall \omega \in L(G)$，ω 称为 GNM 产生的一个语言实例。

设与 GNM 同构的文法模型为 GMM=(V_T,V_N,P,S)，它们之间的关系：$V_{TC} \cup V_{TVF}$ 的功能等价于 V_T,P_{NTF} 功能等价于 $V_N \cup P$。这里 V_{TVF} 的元素具有双重性，即只能作为被调用函数，出现在规则右部；

© 中国科学院软件研究所 http://www.c-s-a.org.cn

又具有像规则一样的动态性。P 集元素之间的关系，反映为 P_{NTF} 集元素 NTF 之间的相互调用、NTF 调用 TC、TVF 的排列和组合上。

GNM 构造主要步骤：

Step1.在目标代码识别实体(元素)、实体的边界，按相似性分类，称为实体(元素)对象。在此基础上，观察元素对象之间的逻辑关系，抽象出逻辑(抽象)对象。分析对象间的相互关系，是否满足规整性。不满足，则不能构造出 GNM。

Step2.所有的实体对象构成 V_T 集，其中凡是变化的元素构成 V_{TV} 集，不变的形成 V_{TC} 集。全部逻辑对象构成 V_N 集。

Step3.逻辑对象，是由元素对象组合而成，体现的是语法规则。逻辑对象，按照功能区分、层级差异，还可构成逻辑对象的逻辑对象，它们统称逻辑对象。逻辑对象之间的关系、逻辑对象跟元素对象的关系构成了 P 集。逻辑的起点，就是 V_N 集的开始符。文法模型 $GMM(V_N,V_T,P,S)$ 构造完成。

Step4.目标语言中事物的数据信息、数据间的关联信息，保存与数据库中，为程序函数所调用。

Step5.终结符变量函数的构造，由非终结符函数传递的数据信息构成函数的参数，返回值是目标语言的一个实体对象。构造出 Step2.V_{TV} 集中的所有终结符变量的函数,形成 VTVF 集。

Step6.GMM 中 V_N 集每一个元素对应一个 NTF；P 集的每一个规则左部同样体现为一个 NTF，右部组成了 NTF 的调用对象。P 集规则描述的信息是模糊的，例如用“*”、“+”等描述，需根据事物节点信息转化为 NTF 内部的逻辑判断或循环语句。所有的 NTF 构成 PNTF 集。

Step7.从数据库中获取数据信息作为参数，传入起点函数 S。按照定义 9，模型 $GMM(V_{TC},V_{TVF},P_{NTF},S)$ 构造完毕。

建立在文法理论基础上的 GNM 是对复杂程序算法进行分析、描述的强力工具。但目标语言的多样性，导致词汇表、规则集的不确定性，如何求出 V、P 是一大难点[7]。

3 树形结构模型

第 1 节描述可知，树形结构满足规整性要求，可按照上文的构造步骤设计生成模型 GNM。

3.1 构造文法模型

字母表 Σ={0-9, a-z, A-Z, <, >, ε, /, ., ', ", _, =, :, (,)}。

终结符集 VT={<table>, </table>, <tr>, <tr...>,</tr>,<td>,<td width=16 height=16>, </td>, , DATA1, DATA2}。(注：源语言部分用省略号表示)

非终结符集 VN={S,L,M,N,O,Q,R,T,U}，开始符为 S。

规则集 P={S--><table>L</table>, L-->(M|N)+,M-->OP,N--><tr><td>DATA1</td></tr>,0--><tr>Q</tr>,P--><tr...>R</tr>,Q-->TU,R--><td></td><td>S</td>,T--><td width=16 height=16><img.../></td>, U--><td>DATA2</td>}

按照源语言示例及文法定义构造的文法模型 GMM(VN,VT,P,S)的语法树，如图 2 所示：

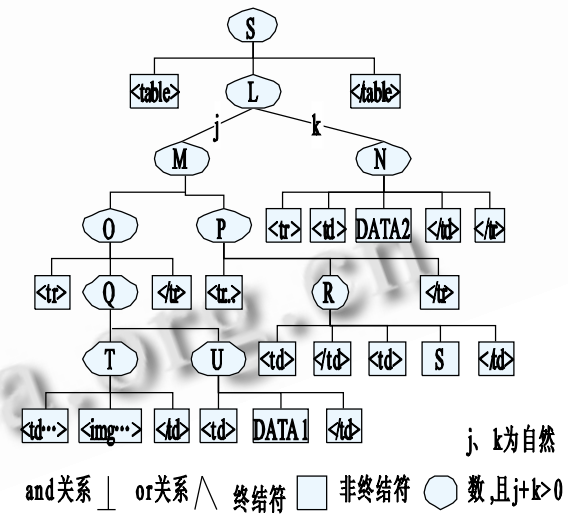

图 2　文法模型语法树

图中 DATA1、DATA2 指树形结构呈现的数据信息，例如：“M27 卡套组件”、“表盖”。j、k 表示对象出现的次数。GMM 语法树可以直观的展现终结符(实体对象)、非终结符(抽象对象)之间的层次、组成关系静态结构，但是无法体现语言生成的自动性。

3.2 构造生成模型

GNM 需要事物数据信息支持。父子节点的关系体现为目的语言的组成关系，即非终结符函数的调用关系。

© 中国科学院软件研究所 http://www.c-s-a.org.cn

终结符常量集 V_{TC}={a,b,c,d,e,f,g,h,i}，其中 a=<table>,b=</table>,c=<tr>,d=e=</tr>,f=<td>,g=i=</td>,h=<td width=16 height=16>。

终结符变量函数集 VTVF={N,U,V,W,X}。非终结符函数集 VNTF={S,L,M,O,P,Q,R,T}，起点 S。

函数之间以及函数对终结符的调用关系结构，如图 3 所示：

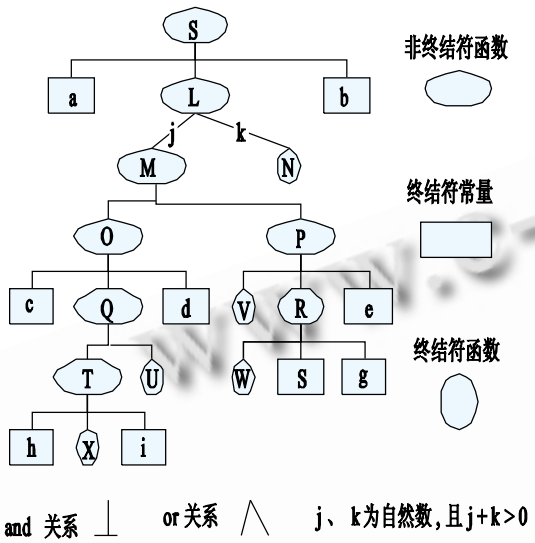

图 3　调用关系图

V_{NTF} 中函数命名可任意，这里为了体现跟 GMM 的 VN 集对应关系，取相同的名字。起点 S 接受数据信息的输入。通过文法理论和函数理论结合，构成自动生成模型 GNM。

3.3 构造函数

GMM、GNM 构造成功，VT、VN、VTC、VTV 明确，则集合每个元素的功能单一、界限明确，终结符变量函数、非终结符函数构造就相对简单。终结符常量集 VTC 在 4.2 节已经构造，下面两种函数各示例一个：

$S \in V_{NTF}$,反应规则 S-->aLb

```
private String S(String name){
String str=a+L(name)+b;
return str;
}
```

$U \in V_{TVF}$

```
private String U(String name){return f+name+g;}
```

其它函数构造雷同。值得一提的是，在非终结符函数 L 含有对数据节点信息进行判断和循环执行的指令。

4　应用实例

测试环境：Windows XP，客户端为浏览器(IE)，Web 服务器采用 Apache 的 Tomcat 5.5，底层数据库为 SQL Server2000，JDK 版本为 1.5。

树型结构的动态更新是来源于节点信息的变化，只要树型结构能够按节点信息的变化而变化就产生了动态效果[3]。不限使用什么数据库,只要在库表中存入相关节点信息并提供改变这些数据的方法,那么实现动态树的先决条件就具备了。在 SQL Servers 库中建立这样的表，表结构、表中节点信息如表 1、表 2 所示(可根据事物信息，拓展数据关系)：

表 1　数据库表结构

表名	instrumentRelating			
含义	列名	数据类型	长度	主键
行号	id	int	4	是
父节点	parent	varchar	50	否
子结点	child	varchar	50	否

表 2　数据库表中的树节点信息

id	parent	child
1	仪表装配图	测温探头
2	仪表装配图	M27 卡套组件
…	…	…

在电子仪表装配树结构当中，树的根为“仪表装配图”。在表现层页面中除了编辑脚本函数 ShowSub 外，再编写一句程序代码 request.getAttribute("html")，取出生成模型的返回值，生成树形结构的目标语言。经浏览器解释执行，显示效果如图 4、图 5 所示：

图 4　树形结构

© 中国科学院软件研究所 http://www.c-s-a.org.cn

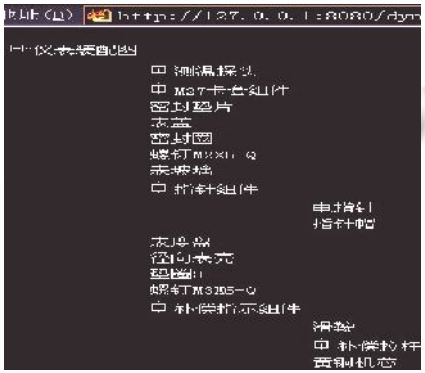

图 5　展开的树形结构

树型结构节点数据存储，采用 Map 二维表的形式，数据库只读取一次，遍历的时间复杂度是 O(n)。

5　总结

对树形结构进行分解和抽象，引出自动生成论域语言的必备条件：组成元素简单，语法规则具递归性；事物数据信息量大，但数据间是层次或组合关系。引入文法理论，扩展终结符含义，构造与非终结符、规则相对应的非终结符函数，提出了分析建模步骤，建立了特定用途语言的自动生成模型。与传统构建树形结构算法相比较，更遵循文法理论规范，更加具有系统性、延展性。生成模型不仅是对文法理论应用的扩展，而且为网页语言自动生成技术提出了一条新的思路。

生成模型应用于众多零件和组件的仪表装配结构中，验证了算法的准确性和实效性。可采取读资源文件的方法，通过改变某些终结符常量值，丰富树形结构的表现效果。

参考文献

1 张步忠,吕强.一个 B/ S 的动态树型结构的设计与实现.苏州大学学报(工科版), 2005,25(3):58－62.

2 储岳中.基于递归算法和树形控件的动态树形图的实现.计算机技术与发展, 2007,17(6):87－89,93.

3 王昕哲,刘万军.在 Web 应用开发中利用静态树和递归算法制作动态树型菜单.计算机与现代化, 2008, 151(3):124－126.

4 孙庆莉,石永杰,张宁.在 ASP.NET 中不使用控件实现动态树形目录.电脑与信息技术, 2004,5(14):55－58.

5 张素琴,吕映芝,蒋维杜,等.编译原理.第二版.北京:清华大学出版社, 2005:33－40.

6 蒋宗礼,姜守恒.形式语言与自动机理论.第二版.北京:清华大学出版社, 2007:48－48.

7 徐兰芳,宋波,吕操,等.由语言自动构造文法的递推描述算法.华中科技大学学报(自然科学版), 2005,33

© 中国科学院软件研究所 http://www.c-s-a.org.cn