ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.8, August 2008, pp.1893−1901
DOI: 10.3724/SP.J.1001.2008.01893
© 2008 by *Journal of Software*. All rights reserved.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

一种基于边的上下文相关图文法形式化框架*

曾晓勤+, 韩秀清, 邹 阳

(河海大学 计算机及信息工程学院,江苏 南京 210098)

An Edge-Based Context-Sensitive Graph Grammar Formalism

ZENG Xiao-Qin+, HAN Xiu-Qing, ZOU Yang

(Computer and Information Engineering College, Hohai University, Nanjing 210098, China)

+ Corresponding author: E-mail: xzeng@hhu.edu.cn

Zeng XQ, Han XQ, Zou Y. An edge-based context-sensitive graph grammar formalism. *Journal of Software*, 2008,19(8):1893−1901. http://www.jos.org.cn/1000-9825/19/1893.htm

Abstract: This paper proposes an edge-based context-sensitive graph grammar formalism with a concentration on solving the main graph grammar problem—embedding problem, and discusses the features of the proposed graph grammar and its parsing algorithm. Some comparisons of the proposed graph grammars with other existing grammars are given. Further researches on graph grammars are also reviewed.
Key words: visual language; formalism; graph grammar; embedding problem; production

摘 要: 围绕解决图文法中的主要问题——嵌入问题,提出了一种基于边的上下文相关图文法形式化框架,并对由此定义的文法的一些性质及相应的归约算法进行了讨论.对所提出的图文法与已有的文法进行了比较.同时,展望了今后值得进一步研究的一些问题和方向.
关键词: 可视化语言;形式化;图文法;嵌入问题;产生式
中图法分类号: TP391 **文献标识码**: A

可视化语言描述的对象通常可以被抽象地看作是结点和边组成的图.在对这类图语言进行定义和分析时,图文法[1,2]是一个有效的形式化工具.图文法是在字符文法的基础上发展起来的,是一维字符串到二维图的自然扩展,主要用来生成图和对图进行分析.目前已有多种图文法以及与之相关的形式化方法问世,并在多个领域得到了应用[3−8],其中最主要的两类是上下文无关图文法和上下文相关图文法.在两类文法中,后者比前者复杂但表达能力强,更适合对可视化语言的形式化描述.现有的上下文有关图文法主要包括 CMG(constraint multiset grammar)[9],PLC(picture layout grammar)[10],LGG(layered graph grammar)[11],RGG(reserved graph grammar)[12]和 SGG(spatial graph grammar)[13]这几种.由于LGG和RGG文法允许产生式左端可以不只是一个结点,而且可以是一个图,因此它们是最具代表性的上下文相关图文法,而且它们也已在可视化语言的定义、分析和变换等方面得到了广泛的应用[14−16].

与字符文法相比,图文法无论是在形式化处理还是在文法类型的划分以及成员问题的判定上都引出了一些新问题,其中最主要的是嵌入问题.这个问题是说,当利用产生式对主图(host graph)的一部分进行替换之后,替

* Supported by the National Natural Science Foundation of China under Grant Nos.60571048, 60673186 (国家自然科学基金)
Received 2008-01-08; Accepted 2008-04-18

© 中国科学院软件研究所 http://www.jos.org.cn

换的部分如何嵌入到主图之中而不产生悬边.在上下文相关图文法中,常见的处理方法是在产生式的两端分别指定一些图的元素并赋予它们之间一种对应关系,然后利用这一对应关系来实现替换部分的嵌入.一般情况下,这些对应的元素是作为产生式的上下文部分.在 LGG 文法中,这些元素是同时存在于产生式两端的一些结点.RGG 则定义了双层结点结构,引入了称为标记(mark)的元素,并要求文法中产生式两端的对应元素是具有相同标记的顶点.这两种文法都利用图中一些元素的对应关系解决了嵌入问题,但还都存在一些不足之处.LGG 将所有的上下文结点都拉入产生式中,这就使得产生式的结构比较复杂;另外,由于在面临不同的上下文结点时要使用不同的产生式才能进行表达,因此,如果为每一种上下文结点都单独定义一条产生式,则会导致产生式的数量很大.LGG 引入的通配符可在一定程度上解决问题,但在很多情况下,要实现原有产生式所表达的语义,所需要使用的通配符数量也可能是很大的,这并不能从根本上解决这一问题.同时,LGG 所给出的归约算法的复杂性很高,在实际问题中很难加以应用.RGG 由于定义的是双层结点的结构,因此看起来不直观,更重要的是,在利用这一文法对图进行变换时,需要先将一般意义下的图转换成 RGG 定义下的图,这一转换过程中包含了如何设置双层结点以及如何将连到该结点上的边分配到各个小结点上等问题.

本文提出了一种基于边的形式化方法来定义上下文相关图文法,称为 EGG(edged graph grammar).提出 EGG 形式化框架的出发点之一是让产生式能够有效地表达抽象的上下文结构信息,而尽量少涉及具体的上下文语义信息.我们认为这样定义的文法抽象程度高、简单且直观,也有利于进一步对图文法自动生成的研究.EGG 承袭了 LGG 和 RGG 在产生式两端定义一些图的元素,并利用这些元素建立对应关系来解决嵌入问题的思想,但这种元素仅使用边(结构信息)而略去了边的一端的上下文结点(语义信息).从检索文献可知,之前也有人在图文法中借用边来处理问题,如 HRG(超边替代图文法)[1]和 StarGrammar[17]中有一些类似的工作.但这两种文法并不是通过两组边之间的对应关系来解决图文法面临的嵌入问题,而且它们的产生式左端本质上都是以一个非终结点为中心的,因而不是典型的上下文相关图文法.

对典型的上下文相关图文法 LGG 和 RGG 而言,EGG 克服了它们中存在的一些不足,展现出了一些较好的特性.与 LGG 相比,这种基于边的上下文相关图文法的产生式结构更加简单,至少省去了表示上下文的结点,不用通配符也可减少产生式的数量,而且当设计的产生式满足“选择无关(selection-free)”条件时就可以使用不需要进行回溯的 SFPA(selection-free parsing algorithm)算法[12],这一算法的复杂性是多项式级的,比 LGG 给出的复杂性为指数级的算法[18]简单了很多.与 RGG 相比,我们所给出的图的结构仍然是一般意义下的图,在使用这一文法对图进行分析和变换时更加方便、直观,而不需要进行图格式上的转换.另外,与 LGG 和 RGG 都不同,EGG 在保证文法是可归约的这一问题上取消了对文法产生式复杂的分层要求,对产生式所作的限制更加直观和宽松,这一方面使文法更易于理解,另一方面也使得产生式易于设计,为进一步研究文法设计自动化打下了基础.

第 1 节在一系列定义的基础上给出基于边的图文法形式化框架,其中包括 EGG 的定义、与 EGG 相关的文法操作的定义、EGG 语言的定义等,并且讨论 EGG 的特性.第 2 节给出 EGG 语言的归约分析算法.第 3 节结合具体实例对 EGG 与已有文法进行比较,主要比较对象是 LGG 和 RGG.第 4 节是全文的总结,并指出今后的研究方向.

1 基于边的上下文相关图文法形式化框架

1.1 基本定义

在对图文法进行形式化定义时,通常是利用集合来进行描述.LGG 和 RGG 均如此,本文也采用这一方法进行相关的定义.如前所述,本文提出的上下文相关图文法是使用边作为产生式两端的对应元素,指出上下文信息.为了实现这一目的,在构建文法时引入了悬边的概念,所谓悬边,是指有一端没有落到任何结点上的边.本文中,把所讨论的图从一般意义下的图拓广到可能带有悬边的图,称为悬边图.下面给出相关的形式化定义.

定义 1.1. $G=(V,E,l,s,t)$是标号集 L 上的一个图.其中:V 是图的结点集,由终结点集 V_T 和非终结点集 V_N 两部分组成;E 是图的边集;$l:V\to L$ 是图中结点的标号函数,指出了每个结点的标号;$s:E\to V$ 和 $t:E\to V$ 分别指明了图中每一条边的起点和终点.

© 中国科学院软件研究所 http://www.jos.org.cn

为方便起见,我们用符号 $G.V,G.E,G.l,G.s$ 和 $G.t$ 分别表示一个图的各个部分,下面的定义都采用这一记法.定义 1.1 给出的是一个一般意义下的图,不含悬边.图中的结点带有标号,边带有方向而不带标号.下面定义的是可以含有悬边的图.

定义 1.2. $Q=(V,\hat{E},M,l,s,t,m)$ 称作标号集 L 上的一个悬边图.其中:V 是悬边图的结点集,由终结点集 V_T 和非终结点集 V_N 两部分组成;$\hat{E}$ 是悬边图的边集,它包含悬边集 $\bar{E}$ 和非悬边集 E,$\bar{E}$ 又分为起点悬边集 $\vec{E}$ 和终点悬边集 $\overleftarrow{E}$ 两部分,两者分别代表起点没有指定的边和终点没有指定的边;M 是悬边图中悬边的标记集;$l:V\to L$ 是结点的标号函数;$s:(E+\vec{E})\to V$ 和 $t:(E+\overleftarrow{E})\to V$ 是边的起点函数和终点函数;$m:\bar{E}\leftrightarrow M$ 是一个双射,称作悬边图的标记函数.

在定义 1.2 中,悬边集 $\bar{E}$ 允许是空集,此时与之相关的标记集 M 也就是空集,而标记函数 m 不再有意义.因此,定义 1.1 描述的图可以看作是定义 1.2 描述的悬边图的一个特例.下文中,在不特别指明时所提到的图都是不带悬边的图,悬边图则包含了带悬边的和不带悬边的图;同样,在不特别指出是悬边时提到的边都指的是非悬边.对于一个悬边图,把除掉它的悬边集之后得到的图称作它的核图,记作 $K(Q)$.把与悬边有关联的结点称作悬边图的关联结点.对于每个关联结点,把以它为起点的悬边的个数称作它的关联出度,记为 $d_{rout}(v)$,即 $d_{rout}(v)=\left|\{\bar{e}\mid \bar{e}\in\bar{E}\wedge s(\bar{e})=v\}\right|$;把以它为终点的悬边的个数称作它的关联入度,记为 $d_{rin}(v)$,即 $d_{rin}(v)=\left|\{\bar{e}\mid \bar{e}\in\bar{E}\wedge t(\bar{e})=v\}\right|$.另外,为了描述方便,把与悬边没有关联的结点的关联出度和关联入度都记为 0.在上面定义的基础上,下面给出基于边的图文法中产生式的定义.

定义 1.3. 由两个悬边图 Q_L 和 Q_R 通过符号“:=”连接而得到的形如 $Q_L:=Q_R$ 的一个结构是一个产生式,其中 $Q_L.M=Q_R.M.Q_L$ 和 Q_R 分别称作产生式的左端和右端.特别地,产生式的左端也可以是文法的初始图,此时产生式的右端必须是不带悬边的图.

如图 1 所示即为一组产生式.

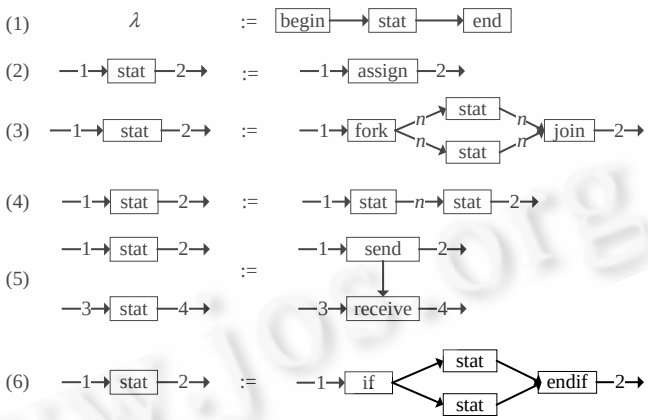


Fig.1 A set of EGG productions

图 1 一组 EGG 产生式

产生式的作用是对图进行变换,这种变换需要基于一定的条件.其中同构是最基本的条件之一.同构的意思是说,在两个图的结点和边之间存在一一对应关系,对应结点具有相同的标号,并且对应边的起点和端点恰好是对应结点.其形式化定义如下:

定义 1.4. 对两个图 $G=(V,E,l,s,t)$ 和 $G'=(V',E',l's',t')$ 来说,如果存在两个双射 $h_V:V\leftrightarrow V'$ 和 $h_E:E\leftrightarrow E'$ 满足下面两个条件,则称 G 和 G' 两图同构,记作 $G\approx G'$:

- $\forall v\in V: l'(h_V(v))=l(v)$;
- $\forall e\in E: s'(h_E(e))=h_V(s(e))\wedge\forall e\in E: t'(h_E(e))=h_V(t(e))$.

在图文法中,子图的定义与图论中有所不同,它除了要求结点和边必须满足包含关系之外,还附加了另外一

© 中国科学院软件研究所 http://www.jos.org.cn

个条件:如果两个结点都包含在子图中,那么端点落在这两个结点上的边必须也在子图当中.

定义 1.5. $G=(V,E,l,s,t)$是一个图,如果图 $G_1=(V_1,E_1,l_1,s_1,t_1)$ 满足下面的条件,就把 G_1 称作 G 的子图:

- $V_1 \subseteq V \wedge \forall v \in V_1(l_1(v)=l(v))$;
- $E_1=\{e \mid e \in E \wedge s(e) \in V_1 \wedge t(e) \in V_1\} \wedge \forall e \in E_1(s_1(e)=s(e) \wedge t_1(e)=t(e))$.

此时,对于 V_1 中的每个结点 V,把从该结点连接到子图外的结点的边数称作它的连通出度,记作 $d_{out}(v)$,即 $d_{out}(v)=\left|\{e \mid s(e)=v \wedge t(e) \in V-V_1\}\right|$; 把从子图外的结点连接到该结点的边数称作它的连通入度,记作 $d_{in}(v)$,即 $d_{in}(v)=\left|\{e \mid t(e)=v \wedge s(e) \in V-V_1\}\right|$. 把 G 中两个端点分别落在 V_1 和 $V-V_1$ 上的边称作 G_1 在 G 中的桥边,记为 $E(G_1 \backslash G)$.

在定义了同构和子图的概念之后,就可以在一个图中寻找满足替换条件的子图并借助产生式对其进行替换,我们把这样的子图称作图柄(redex),其定义如下:

定义 1.6. X 是图 G 的一个子图,Q 是一个悬边图;如果 X 与 Q 的核图 $K(Q)$同构,并且对于同构函数下的任意一组对应结点 $v \in X.V$ 和 $v' \in Q.V$, 都满足 $d_{out}(v)=d_{out}(v')$和 $d_{in}(v)=d_{in}(v')$,则把 X 称作悬边图 Q 的一个图柄.

对于一个图柄 X,一定能够在 $E(X \backslash G)$和 $Q.\overline{E}$ 所包含的边之间构建一个双射 $g: E(X \backslash G) \leftrightarrow Q.\overline{E}$,使对应边在 X 和 Q 中的起点(终点)恰好也是 X 和 $K(Q)$的同构函数下的对应结点.我们把 g 称作图柄所决定的边的映射.特别地,当 Q 不含悬边时,此映射无定义.这并不会影响到图的变换过程,下文中将不单独进行说明.在一个给定图中找到图柄之后,就可以借助产生式对图柄进行替换,从而得到一个新图,这一过程称作图的变换.其定义和步骤如下:

定义 1.7. 如果一个图 G 的子图 X 是悬边图 Q_L 的图柄,而 Q_L 和 Q_R 分别是一个产生式 p 的左端和右端,则可以用 Q_R 代替 X 在 G 中的位置而得到一个新图 G',步骤如下:

1. 从图 G 中删除 X 及所有有端点落在 X 中结点上的边;
2. 将 Q_R 添加到当前图中:对于 Q_R 中的每条悬边 $\overline{e}$, 假设 Q_L 中与其标记相同的边为 $\overline{e}'$,而在图柄决定的边的映射 g 下,$\overline{e}'$对应的 $E(X \backslash G)$中的边为 e'',则将 $\overline{e}$ 连接到 e'' 落在 $G-X$ 中的端点上.

需要指出的是,当 Q_R 中的一个关联结点同时连接多于 1 条悬边时,有可能无法唯一确定这几条边分别和 X 对应结点上所连接的哪一条桥边相对应,或者说图柄 X 所决定的边的映射会不止一种(是各种可能映射的组合数),此时应视不同映射的图柄 X 为不同的图柄,即用同一个产生式对图柄进行替换,可能会由于选择的映射关系不同而在替换之后得到不同的图.此后,在不特别指明的情况下,下文中所说的图柄都是指已经确定了映射关系的,也就是说,对应不同映射关系的同一个图柄都被看作不同的图柄.

上面描述的是在主图中寻找产生式左端的图柄并用右端进行替换的过程,这一操作称作图的推导;反之,也可以寻找产生式右端的图柄并用左端进行替换,这称作图的归约,其过程与推导类似.在不严格区分推导与归约时,将其统称为图的变换.为了便于描述,我们引入几个记号.如果图 G 经过一步变换可以得到 G',则记作 $G \Rightarrow G'$; 如果图 G 经过 n 步变换($n \le 0$)可以得到 G',则记作 $G \Rightarrow^* G'$. 为了区分推导和归约,在推导时将这两个符号记作 $\mapsto$ 和 $\mapsto^*$, 归约时将这两个符号记作$\rightarrow$和$\rightarrow^*$.图 2 所示为利用图 1 的产生式对初始图进行推导的过程.

1.2 基于边的上下文相关图文法的定义和性质

在有了如前所述的定义之后,就可以对基于边的图文法 EGG 及其产生的语言进行定义.

定义 1.8. 一个 EGG 是一个四元组(A,T,N,P).其中,A 是文法的初始图;T 和 N 分别是终结标号集和非终结标号集,两者的并集 $T \cup N$ 是标号集 L;P 是文法的产生式集,它所包含的每一个产生式 $p=Q_L:=Q_R$,首先要满足定义 1.3 的形式要求,此外还必须满足下面的条件:

(1) Q_L 和 Q_R 的标号集都必须是 Q 的子集;

(2) $Q_L<Q_R$.其中“<”定义的是两个悬边图之间的关系,具体定义如下:

$$Q<Q' \Leftrightarrow (|Q.V| \lhd |Q'.V|) \vee (|Q.V|=|Q'.V| \wedge |Q.V_T| \lhd |Q'.V_T|) \vee (|Q.V|=|Q'.V| \wedge |Q.V_T|=|Q'.V_T| \wedge |Q.\hat{E}| \lhd |Q'.\hat{E}|).$$

定义 1.9. 如果(A,T,N,P)是一个 EGG,则将 $\{G \mid A \mapsto^* G \wedge |G.V_N|=0\}$ 称作 EGG 定义的语言.

© 中国科学院软件研究所 http://www.jos.org.cn

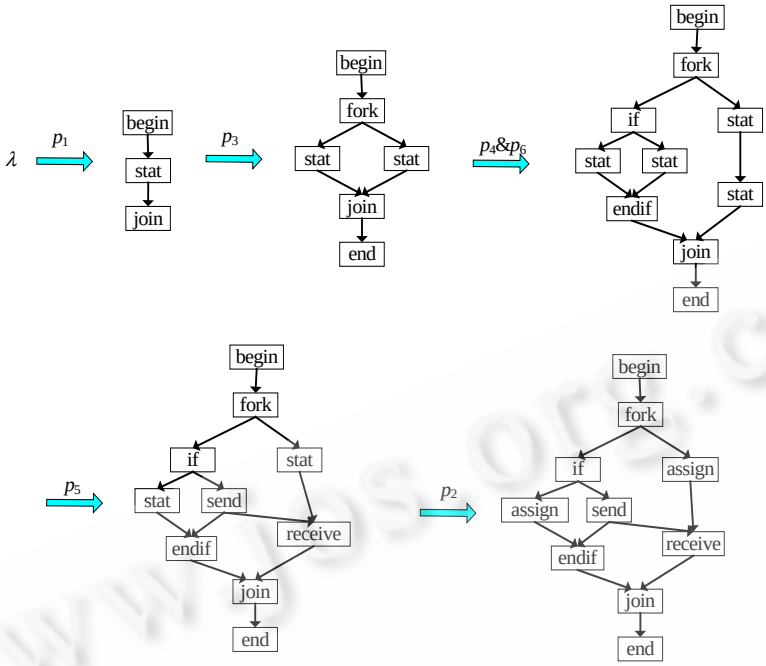


Fig.2 A graph derivation process using EGG

图 2 一个用 EGG 推导图的过程

图文法不仅可以用来定义图的语言,而且还是对图进行分析的依据,主要是判定一个图是否属于给定图文法所定义的语言,也就是根据文法的产生式通过对所给图进行归约来判定.如果一个图可以归约到初始图,就说明这个图属于图文法定义的语言;反之,则说明它不属于图文法定义的语言.这一命题对 EGG 也是成立的.下面对此进行证明.

引理 1.1. 对于 EGG,如果 $G \mapsto G'$,那么 $G' \to G$.

证明:假设图 G 在利用产生式 $Q_L{:=}Q_R$ 进行推导后得到 G',对应的图柄为 X,那么对于图 G'来说,由 Q_R 替换入的部分组成的子图一定与 Q_R 的核图同构,且对应结点在 G'中的连通出度(连通入度)与在 Q_R 中的关联出度(关联入度)是相等的.也就是说,它是图 G'中相对于 Q_R 的一个图柄,因此可以用产生式 $Q_L{:=}Q_R$ 对 G'进行归约.由嵌入方法可知,此过程完成之后得到的图一定是图 G,即 $G' \to G$ 得证. □

引理 1.2. 对于 EGG,如果 $G' \to G$, 那么 $G \mapsto G'$.

证明与引理 2.1 的证明类似.

定理 1.1. 若(A,T,N,P)是一个 EGG,如果 $A \mapsto {*G}$, 那么 $G \to {*A}$.

证明:由 $A \mapsto {*G}$ 可知存在一个推导过程如下: $A \mapsto G_1 \mapsto G_2 \mapsto G_3 \mapsto \ldots \mapsto G_{n-1} \mapsto G$. 由引理 1 可知,下述过程也是成立的: $G \to G_{n-1} \to \ldots \to G_3 \to G_2 \to G_1 \to G$. 于是, $G \to {*A}$ 得证. □

定理 1.2. 若(A,T,N,P)是一个 EGG,如果 $G \to {*A}$,那么 $A \mapsto {*G}$.

证明与定理 1.1 的证明类似.

上面证明了一个图属于图文法产生的语言和该图可以归约到文法的初始图两者是等价的.下面将证明,对于任意一个图,都可以判定其是否属于 EGG 定义的语言.

定理 1.3. 给定一个具体的 EGG,对任意一个图都可以判定其是否属于该图文法的语言.

证明:通过对图文法归约过程的分析对此加以证明.假设 G 是给定的任意图,A 是文法的初始图,对 G 进行归约,可以得到如下序列: $G, H_1, H_2, \ldots, H_{n-1}, H_n$, 其中 $G \to H_1, H_i \to H_{i+1} (1 \le i \le n-1)$, 且 H_n 已不能继续进行归约.由 EGG 对产生式的限制条件 2 可知,该序列中任意一项不会再归约得到它前面的某一项,也就是说,不会出现循环

© 中国科学院软件研究所 http://www.jos.org.cn

的情况;同时,由于该序列中右边的图都是“<”左边的,而图 G 的结点数、终结点数、边都是有限的,因此归约过程必可在有限步内完成.由于每一步得到的都是有限图,其可能的图柄也都是有限的,因此一定能够穷举出所有这样的序列.所以,如果在这些序列得到的最终图中包含了初始图,则说明图 G 属于该图文法产生的语言;否则,说明它不属于该图文法产生的语言.证毕. □

2 基于边的上下文相关图文法的归约

如上一节的证明所示,可以通过穷举所有的归约路径来判定某个图是否属于给定的图文法所产生的语言.在具体操作时,可以按照下述步骤进行:

1. 以所给图为起点不断地在当前图中寻找图柄并进行归约,一直到找不到当前图的图柄为止.此时,若得到的是文法的初始图,则说明该图属于图文法产生的语言,判定完毕.否则,转入步骤 2.
2. 将当前图向前回溯一步,然后寻找另外的图柄,如果找不到新的图柄则继续向前回溯,直到可以找到另外的图柄为止.若已经回溯到所给图,且无法再找到新的图柄,则说明该图不属于图文法产生的语言,判定完毕.否则,在找到新的图柄之后转入步骤 3.
3. 利用步骤 2 中新找到的图柄对当前图进行变换,并以此时得到的图为起点进行步骤 1 的操作过程,即不断进行归约,直到找不到图柄为止.此时,若得到文法的初始图说明该图属于图文法产生的语言,则判定完毕.否则,转入步骤 2.

使用这一算法可以判定一个图是否属于给定的图文法.但这一算法的复杂性较高,在最坏情况下需要穷尽所有的可能才能完成判定.在 RGG 文法中引入了 SFPA 算法[12],这一归约算法不需要进行回溯,只要不断为当前图寻找图柄进行替换,直到不能继续下去为止即可.也就是说,只需要进行上述 3 个步骤中的第 1 步,如果此时得到初始图说明给定图属于图文法产生的语言;否则,说明给定图不属于图文法产生的语言,两种情况下判定都在此时结束.

SFPA 算法的使用是有条件的,文献[12]中给出了 RGG 文法使用这一算法的条件.我们针对 EGG 文法的形式,对这一条件进行了改写.

定义 2.1. P 是一个 EGG 的产生式集,如果其中的任意两个产生式 p_1 和 p_2,对它们右端 Q_1 和 Q_2 的任意一个融合 Q 而言,只要 Q_1 和 Q_2 都是 Q 的图柄,那么先用 p_1 后用 p_2 对 Q 进行变换和先用 p_2 后用 p_1 对 Q 进行变换,效果都是相同的,即 $\exists Q_a,Q_b,Q_{ab},Q_{ba}(Q \to^{p_1} Q_a \to^{p_2} Q_{ab} \wedge Q \to^{p_2} Q_b \to^{p_1} Q_{ba} \wedge Q_{ab} \approx Q_{ba})$, 此时,称文法的产生式集是选择无关的.

上述定义中融合的概念可参见文献[12].只要一个 EGG 文法的产生式集满足“选择无关”条件,就可以使用 SFPA 算法,从而大大降低了归约的复杂性.但我们注意到,这一条件对 EGG 来说限制过强,其主要原因在于,EGG 文法中可能会出现主图中的一个子图被视为几个不同图柄的情况.下一步我们将针对 EGG 文法探索降低归约算法复杂性的方法,目前多数情况下还只能使用带回溯的归约算法.

3 与已有文法的比较

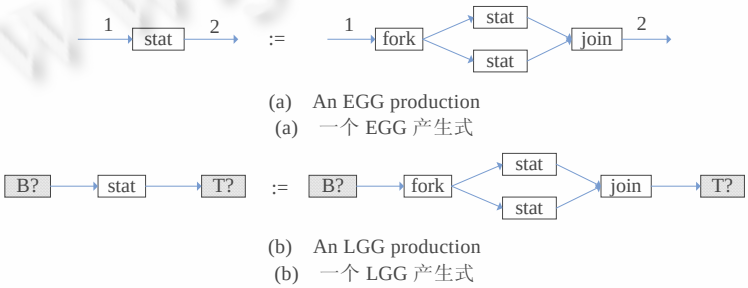
多数的图文法除了要定义产生式规则以外都还要定义一组嵌入规则来解决嵌入问题,而它们的嵌入规则或多或少都要涉及到语义信息,既难以形式化表达又难以理解,与引入形式化方法的初衷是相悖的.EGG 作为一种上下文相关图文法,利用图的边来表示上下文信息,用极为简单的嵌入规则来解决推导和归约中可能产生悬边的问题,避免了使用掺杂语义的非形式化的嵌入规则.

在 HRG[1]和 StarGrammar[15]中也借用边来解决图文法中的一些问题.但前者的产生式左端是唯一的一个非终结点加上若干条用来与其他结点相连接的触角,产生式右端与这些触角对应的是结点而不是边,而且也不能描述在产生式右端的一个结点上同时伸出多条边的情形.后者的产生式左端则是以一个结点为中心将与其相邻的结点同时带入到产生式中,因此其结构有很大的局限性,而且实现嵌入时所依赖的也不是边的对应关系.

EGG 与 LGG 和 RGG 有诸多相似之处.它们都是在产生式的两端放置了一些对应的图的元素,通过这些元

© 中国科学院软件研究所 http://www.jos.org.cn

素之间的对应关系来解决嵌入问题.与前两种文法不同的是,EGG 选取边而不是结点来作为这种元素,这使得 EGG 较之 LGG 有更为灵活的表达方式,较之 RGG 有更简洁的表现形式.与 LGG 相比,EGG 的产生式结构更加简单、直观,避免了使用通配符要涉及的语义处理,所需要的产生式数量也可最大限度地减少,如图 3 所示,图 3(a)和图 3(b)分别是 EGG 的产生式和 LGG 的产生式.图 3(a)表达的是把一个 stat 结点(有一条出边和一条入边)替换成右端的结构,同时指明了原本连接在 stat 上的边在替换之后连接到何处.对于这个产生式来说,并没有明显的上下文结点,也就是说,无论这两条边在主图中所连接的是什么样的结点,都可以进行这种替换.如果要用 LGG 文法来表达,则必须先穷举出所有可能的上下文结点,然后通过定义通配符试图减少产生式数量.如图 3(b)所示,B?和 T?就是两个通配符,将 B?定义为{begin,fork,if,stat},将 T?定义为{end,assign,join,send,receive},当一个结点的标号是相应集合中的元素时,就可以与其匹配.这种方法虽然有可能减少产生式的数量,但多出了对通配符的处理,而两个不同的产生式所需要的上下文集合只要有一个元素不同,就要设置不同的通配符,因此通配符的数量可能会很大,导致文法产生式的构造复杂化.从归约算法上分析,由于 EGG 的产生式结构比 LGG 要简单,因此在寻找图柄时更为简单.另一方面,由于产生式较为简单,这意味着对图柄的限制比较少,可能找到的图柄比 LGG 要多,通常这会导致回溯次数的增多,但由于该文法在产生式满足“选择无关”条件时,可以使用无须回溯的 SFPA 算法,此时,这个问题就不会影响到整个归约的复杂性.总体上讲,当可以使用 SFPA 算法时,EGG 的归约复杂性要比 LGG 降低很多,达到了多项式级;而在不满足使用 SFPA 算法的情况下,两种文法的归约复杂性都是指数级的.



(a) An EGG production
(a) 一个 EGG 产生式

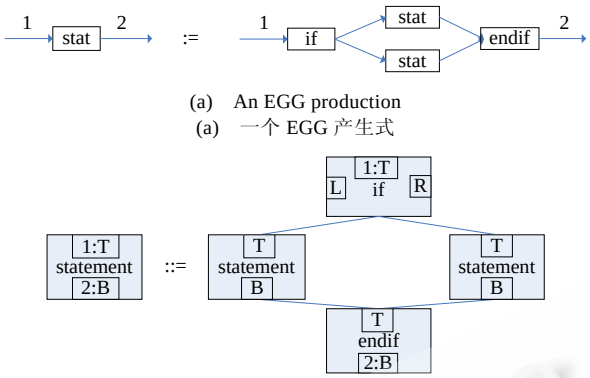
(b) An LGG production
(b) 一个 LGG 产生式

Fig.3 A production comparison between EGG and LGG
图 3 EGG 和 LGG 产生式的比较

与 RGG 文法相比,EGG 的优点主要体现在图的结构上.如前所述,RGG 由于定义了双层结点结构,在实际应用时存在这种图和一般意义下的图的转换问题.对于一般的使用者来说,要实现这种转换并不容易.而 EGG 所定义的图是一般意义上的图,产生式中的图虽和一般的图有所不同,但仍然是一般意义下图的一部分,在运用到图的变换过程中的时候,根据嵌入规则可以很容易地嵌入到主图当中.如图 4 所示,是表达意义相同的两个产生式,前者属于 EGG,后者则属于 RGG.这两个产生式表达的语义是类似的,但结构上前一产生式要比后一产生式简单.更为重要的是,在利用后一产生式对图进行变换时,还要将主图转换成和产生式中的图相同的结构之后才能够判定某个子图是否是一个图柄,而在转换过程中如何设置双层结点以及如何将连到该结点上的边分配到各个小结点上等并不是一件简单的事情.

另外,EGG 和 LGG 和 RGG 都不同的一点还体现在为了保证文法的可归约性而对产生式所作的限制上.LGG 和 RGG 都是通过对结点标号进行分层,并基于这种层次关系对产生式作限制的,这种方法比较复杂,也不直观.而 EGG 则借用了 RGG+[19]直接利用终结点和非终结点的数目来对产生式进行限制的方法,这使文法看起来更加直观和易于理解,同时还加入了边的元素,从而使得对产生式所作的限制更加宽松,文法设计更加自由.

© 中国科学院软件研究所 http://www.jos.org.cn



(a) An EGG production
(a) 一个 EGG 产生式

(b) An RGG production
(b) 一个 RGG 产生式

Fig.4 A production comparison between EGG and RGG
图 4 EGG 和 RGG 产生式的比较

4 总结与展望

我们的研究目标是在试图保持上下文有关文法表达能力的前提下简化文法的结构和产生式的规则,为文法自动生成的实现打下基础.本文提出的 EGG 形式化方法就是着眼于用尽可能简单的产生式规则借助多次推导或归约来处理复杂的图语言.EGG 简化了表示,但直觉上它的表达能力较之 LGG 和 RGG 并没有变弱.因为 EGG 只是省去了 LGG 产生式中的上下文结点,而依附结点的边依然保留,实质上是保留了结构信息而丢掉了一些语义信息(结点的标号);再有,EGG 的悬边加标记起的作用可与 RGG 的小结点加标记的作用相类似.EGG 简化表示的代价是有可能增加推导或归约过程的复杂性,以归约为例,用结构简单的产生式右图在主图中寻找句柄,可能会导致句柄数的增多(如前文提到的要考虑组合情况),从而引起语法分析过程中回溯的增加.但是,从另一方面来看,用结构简单的图在主图中寻找句柄时,有可能降低图匹配过程的复杂性.

本文提出了一种基于边的上下文相关图文法的形式化框架.与已有的图文法形式化方法相比,这种方法具有以下几个优点:

① 定义的图文法抽象程度高,不涉及复杂的语义信息,嵌入规则和文法结构简单、直观;

② 对产生式的限制少,规则简化,易于理解和设计;

③ 产生式的简化并未削弱文法表达能力,借助形式化机制(推导和归约等)仍可处理较为复杂的图;

④ 文法满足“选择无关”条件时,可以使用复杂性较低的 SFPA 算法.

接下来,我们将在理论上深入研究 EGG 与 LGG 和 RGG 之间在表达能力上的关系.为使 EGG 具有实用性,我们将从多方面寻求降低 EGG 语法分析复杂性的途经,包括诸如子图的匹配、句柄的确定、产生式的选择、回溯的减少、子图的替换等.此外,我们还将构建 EGG 的相应支撑软件,考虑 EGG 在可视语言领域里的应用,并研究图文法的自动生成问题.

References:

[1] Rozenberg G. Handbook of Graph Grammars and Computing by Graph Transformation, Volume 1. Singapore: World Scientific Publishing Co. Pte. Ltd., 1997.

[2] Fahmy H, Blostein D. A survey of graph grammars: Theory and applications. In: Proc. of the 11th Int’l Conf. on Pattern Recognition, Vol.II: Conf. B: Pattern Recognition Methodology and Systems. 1992. 294−299.

[3] Fan L, Xie L. Relation graph grammar and its application. Journal of Software, 1997,8(2):87−92 (in Chinese with English abstract).

[4] Xu JL, Zhou LX. A graph grammar based concurrent system testing method and its implementation. Journal of Software, 1996, 7(10):587−605 (in Chinese with English abstract).

© 中国科学院软件研究所 http://www.jos.org.cn

[5] Blostein D, Schürr A. Visual modeling and programming with graph transformations. In: Proc. of the 14th IEEE Symp. on Visual Languages. Los Alamitos: IEEE Computer Society Press, 1998.

[6] Ma XX, Cao JN, Lü J. Architecting distributed Web applications: A graph-oriented approach. Chinese Journal of Computers, 2003, 26(9):1104−1115 (in Chinese with English abstract).

[7] Taentzer G, Carughi GT. A graph-based approach to transform XML documents. In: Baresi L, Heckel R, eds. Proc. of the FASE 2006. LNCS 3922, Berlin, Heidelberg: Spring-Verlag, 2006. 48−62.

[8] Shi B, Ran P, Ma XX, Tao XP, Lü J. Attributed graph grammar-based description and constraints verification of software architectures. Application Research of Computers, 2007,24(3):163−168 (in Chinese with English abstract).

[9] Marriott K. Constraint multiset grammars. In: Proc. of the 10th IEEE Symp. on Visual Languages. Los Alamitos: IEEE Computer Society Press, 1994. 118−125.

[10] Golin E. A Method for the specification and parsing of visual languages [Ph.D. Thesis]. Department of Computer Science, Brown University, 1991.

[11] Rekers J, Schürr A. Defining and parsing visual languages with layered graph grammars. Journal of Visual Languages and Computing, 1997,8(1):27−55.

[12] Zhang DQ, Zhang K, Cao JN. A context-sensitive graph grammar formalism for the specification of visual languages. The Computer Journal, 2001,44(3):187−200.

[13] Kong J, Zhang K, Zeng XQ. Spatial graph grammars for graphical user interfaces. ACM Trans. on Computer-Human Interaction, 2006,13(2):268−307.

[14] Zhang K, Zhang DQ, Cao JN. Design, Construction, and application of a generic visual language generation environment. IEEE Trans. on Software Engineering, 2001,27(4):289−307.

[15] Zhang K, Zhang DQ, Deng Y. Graphical transformation of multimedia XML documents. Annals of Software Engineering, 2001, 12(1):119−137.

[16] Zhang KB, Orgun MA, Zhang K. A prediction-based visual approach for cluster exploration and cluster validation by HOV^3. In: Kok JN, Koronacki J, de Mantaras RL, Matwin S, Mladenic D, Skowron A, eds. Knowledge Discovery in Databases: Proc. of the 11th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD 2007). 2007. 336−349.

[17] Drewes F, Hoffmann B, Janssens D, Minas M, Eetvelde NV. Adaptive star grammar. In: Corradini A, Ehrig H, Montanari U, Ribeiro L, Rozenberg G, eds. Proc. of the ICGT 2006. Berlin: Springer-Verlag, 2006. 77−91.

[18] Rekers J, Schürr A. A parsing algorithm for context-sensitive graph grammar. Technical Report, 95-05, Leiden University, 1995.

[19] Zeng XQ, Zhang K, Kong J, Song GL. RGG+: An enhancement to the reserved graph grammar formalism. In: Erwig M, Schurr A, eds. Proc. of the 2005 IEEE Symp. on Visual Languages and Human-Centric Computing. Los Alamitos: IEEE Computer Society Press, 2005. 272−274.

附中文参考文献:

[3] 方林,谢立.关系图文法及其应用.软件学报,1997,8(2):87−92.

[4] 徐建礼,周龙骧.基于图文法的并发系统状态测试方法及其实现.软件学报,1996,7(10):587−605.

[6] 马晓星,曹建农,吕建.一种面向图的分布 Web 应用架构技术.计算机学报,2003,26(9):1104−1115.

[8] 石兵,冉平,马晓星,陶先平,吕建.软件体系结构的属性图文法描述及其约束验证.计算机应用研究,2007,24(3):163−168.

曾晓勤(1957－),男,安徽合肥人,博士,教授,博士生导师,CCF 高级会员,主要研究领域为人工神经网络,图文法.

邹阳(1976－),男,讲师,主要研究领域为图文法,非单调推理,逻辑程序.

韩秀清(1981－)男,硕士生,主要研究领域为图文法及其应用.

© 中国科学院软件研究所 http://www.jos.org.cn