

上下文相关图文法的表达能力分析*

邹阳^{1,2+}, 吕建¹, 曹春¹, 胡昊¹, 宋巍¹, 杨启亮^{1,3}

¹(计算机软件新技术国家重点实验室(南京大学), 江苏 南京 210093)

²(河海大学 计算机与信息学院, 江苏 南京 210098)

³(解放军理工大学 工程兵工程学院, 江苏 南京 210007)

On the Expressiveness of Context-Sensitive Graph Grammars

ZOU Yang^{1,2+}, LÜ Jian¹, CAO Chun¹, HU Hao¹, SONG Wei¹, YANG Qi-Liang^{1,3}

¹(State Key Laboratory for Novel Software Technology (Nanjing University), Nanjing 210093, China)

²(Computer and Information College, Hohai University, Nanjing 210098, China)

³(Engineering Institute of Corps of Engineers, PLA University of Science and Technology, Nanjing 210007, China)

+ Corresponding author: E-mail: yzou08@gmail.com

Zou Y, Lü J, Cao C, Hu H, Song W, Yang QL. On the expressiveness of context-sensitive graph grammars. *Journal of Software*, 2012, 23(7): 1635–1655 (in Chinese). <http://www.jos.org.cn/1000-9825/4085.htm>

Abstract: Context-Sensitive graph grammars are formal tools used for specifying visual languages. In order to intuitively describe and parse visual languages, current research has stressed the formalisms and algorithms of graph grammars, but has neglected the comparison of their expressiveness. Based on the analysis and induction of the key characteristics of context-sensitive graph grammar, the relationships between their expressiveness are uncovered and proved in this paper by constructing formalism-transforming algorithms. Moreover, the proposed algorithms correlate with these formalisms; thus, facilitating the usage of context-sensitive graph grammars, as alternative formalisms rather than merely one can be chosen to separately specify and parse visual objects in applications.

Key words: visual language; context-sensitive; graph grammar; expressiveness; transforming algorithm

摘要: 上下文相关图文法是描述可视化语言的形式化工具。为了直观地刻画并高效地分析可视化语言,已有图文法形式框架均着重于文法形式和分析算法的研究,而忽略了对它们之间表达能力的分析。在对已有上下文相关图文法形式框架的关键特征进行分析和归纳的基础上,通过构造不同形式框架之间的转换算法,揭示并形式化证明了它们表达能力之间的关系。而且,转换算法在不同形式框架之间建立了关联,使图文法的应用不必再局限于一个框架,而是可以选择不同框架分别进行图的描述和分析,从而提高了上下文相关图文法的易用性。

关键词: 可视化语言;上下文相关;图文法;表达能力;转换算法

中图法分类号: TP301 文献标识码: A

可视化语言(visual language)是由一组给定语法和语义的可视化句子(图)所构成的集合,主要包括可视化信

* 基金项目: 国家自然科学基金(61021062, 61170089, 61003019); 国家重点基础研究发展计划(973)(2009CB320702); 核高基重大专项(2009ZX01043-001-06); 河海大学中央高校基本科研业务费(2009B21914)

收稿时间: 2010-06-26; 修改时间: 2011-03-29; 定稿时间: 2011-07-01

息处理语言和可视化编程语言^[1].前者常用于图像处理、计算机视觉、办公自动化和图像通信等领域,而后者在计算机图形学、用户界面设计、计算机辅助设计等领域有着十分广泛的应用.对于可视化编程语言,为了建立可自由编辑且独立于特定语言的可视化编程环境,一个通用的可视化语言的形式化定义机制是必不可少的,这是因为语法结构的形式化定义是构造语法分析器的基础.在可视化语言描述方面,早期具有代表性的研究工作有RG(relational grammars)文法^[2]、Golin等人提出的PLG(picture layout grammar)文法^[3]和Marriott的CMG(constraint multiset grammar)文法^[4].它们均采用符号形式的产生式来刻画图元之间的空间关系,照此组装之后生成所定义的图.所以,本质上它们刻画的是空间关系图(spatial relations graph,也称为具体语法图).而且,它们都采用一维的符号串形式来描述二维的图,由于描述的形式和内容之间的不统一,产生式的形式缺乏直观性,可理解性较差.

一类重要的可视化编程语言,如控制与数据流图、状态转换图、Petri网、UML图和ER图等,已广泛应用于计算机理论、软件和数据库等领域.在这些应用环境中,往往需要描述可视化句子的语法结构甚至语义信息,即抽象语法图^[5],所以只配备一个对应的空间关系图是远远不够的.而且,鉴于可视化语言的二维特性,与一维字符串文法(Chomsky文法)相比,采用二维图文法来描述其空间关系图或抽象语法图是一个更为自然的选择.

图文法是定义图语言和对图进行语法分析的形式化工具.一个图文法通常由一个初始图和一组图产生式组成.由于图文法描述的是图元之间的二维连接关系,因而不可避免地引入了嵌入问题,即对一个主图(host graph)应用图产生式进行子图替换后,所生成的新图中可能出现悬边.解决嵌入问题的一般途径是使用嵌入规则,即依附在产生式上用于刻画应用此产生式时替换子图与余图(原图删除被替换子图后所得到的图)之间的连接方法.嵌入规则主要有无限制、方向和标号保持、深度为1和不变型等类型;依照产生式形式分类,图文法主要包括上下文无关和上下文相关文法两大类^[6].而图文法的表达能力则取决于其嵌入问题解决机制和文法所类别.对于一个图文法形式框架而言,表达能力、产生式的形式以及语法分析算法的效率是至关重要的特性.

上下文无关的超图文法(hypergraph grammar)可用于可视化语言的描述^[7],一个超图是指由一个有限结点集与一个有限超边集所构成的有向图.为了增强表达能力,超图文法引入了可用于在上下文图元上生成额外超边的嵌入产生式.尽管如此,超图文法仍然不能表示一些可视化语言,如UML类图等.

Rekers和Schürr^[8]提出了LGG(layered graph grammar)文法形式框架,这是最早用于描述可视化语言的上下文相关图文法.LGG采用在产生式两端同时引入上下文的方法来解决嵌入问题,并使用通配符来刻画同一产生式中不同的上下文结点和边.LGG文法很直观,易于理解,但产生式中出现的上下文元素也导致了产生式结构不够简洁,因而语法分析算法非常复杂,时间复杂度为指数级;而且由于通配符的引入,产生式的设计也较为复杂.在LGG的基础上,Zhang提出了RGG(reserved graph grammar)文法^[9].RGG定义了一种两层的结点结构,一个结点内可以包含若干个顶点,结点本身也是一个超级顶点,所有顶点均可作为与该结点连接的边的端点.基于这种特定结点所构成的图,RGG设计了一种顶点标记机制,使产生式左右两端的被标记顶点一一对应以解决嵌入问题.对于满足合流(confluence)约束的RGG文法,可以构造多项式时间复杂度的语法分析算法.与LGG相比,RGG省去了上下文元素及通配符,产生式的形式更为简洁;但由于引入了特殊的结点结构和标记机制,导致产生式的形式不够直观,文法设计较为困难.为了尝试解决LGG和RGG的不足之处,两种新的文法框架CAGG(context-attributed graph grammar)^[10]和EGG(edged graph grammar)^[11]被分别提了出来.其中,CAGG文法在结点上附加了一种上下文属性,利用结点属性值来刻画可能的上下文信息.通过显式定义产生式左右两端的属性结点之间的对应关系来解决嵌入问题;而EGG文法在产生式两端的有向图上分别附加一组悬边,且左右图上的悬边之间一一对应,利用悬边来描述应用产生式时被替换子图与余图中上下文结点之间的连接关系,以解决嵌入问题.这两种文法都比LGG和RGG更为简洁,但又都不及LGG直观.此外,二者都延续了RGG做法,通过引入合流条件来构造语法分析算法.上述图文法形式框架都是上下文相关的,且都采用的是不变型嵌入规则.它们已经较为广泛地应用于可视化语言等领域,如ER图定义^[8]、过程流图描述^[10]、程序行为发现与验证^[12]、UML图行为语义描述^[13]、模型管理^[14]、设计模式演化^[15]、XML文档设计与转换^[16]、多媒体布局自适应^[17]和图形用户界面设计^[18]等.

但是,在 LGG 之后,其他上下文相关图文法形式框架的提出都是为了尝试解决已有文法框架在产生式形式和语法分析算法上的不足,而忽略了对它们表达能力之间关系的分析和比较,而这恰恰是图文法形式理论不可或缺的内容.而且,已有图文法形式框架在这些特性上各不相同,有的产生式形式直观但较为复杂,且分析算法复杂度极高;有的形式不够直观但较为简洁,且分析算法复杂度较低.一般而言,直观的图文法形式有利于产生式的设计,而简洁的图文法形式则有助于降低语法分析算法的复杂度.但是,由于缺乏这些图文法之间表达能力的理论结果以及不同图文法之间相互转换的方法,在针对某个具体应用选择图文法时,通常只能局限于一种形式框架,而不能在设计图文法实例、生成图例和应用语法分析算法等不同应用场景中扬长避短地选择适合当前情形的不同形式框架,因而导致已有图文法形式框架在实际应用中的效能大打折扣.

归纳起来,不论从理论研究还是从图文法应用的视角来看,上下文相关图文法不同形式框架之间表达能力的分析都是亟待研究的问题.鉴于此,本文在深入剖析已有上下文相关图文法形式框架基本特性的基础上,提出不同形式框架之间的相互转换机制,进而分析并且形式证明其表达能力之间的关系.本文工作的主要贡献在于:(1) 建立了已有上下文相关图文法形式框架表达能力之间的关系,从理论上弥补了在上下文相关图文法形式框架表达能力研究上的不足;(2) 构造了不同上下文相关图文法形式框架的文法实例之间的转换算法,为图文法表达能力的理论结果提供了一条切实可行的应用途径.转换算法构建了上下文相关图文法之间的关联,让图文法的应用不必再拘泥于一个形式框架,而是可以在不同形式框架之间选择合适的描述形式与分析算法,从而使图文法的应用更为自由和方便,而且应用的效率也更高.

本文第 1 节简要介绍已有的 4 种上下文相关图文法的形式框架,进而分析和比较它们的基本特征.第 2 节给出图文法表达能力的定义,而且在构造转换算法的基础上证明不同图文法之间表达能力的关系.第 3 节对相关工作进行比较.最后,第 4 节总结本文的工作.

1 上下文相关图文法

一个图文法 (A, P) 由一个初始图 A 和一组图产生式 P 构成,常用于一类相关图的定义及其语法结构分析.而一个图产生式 $p \in P$ 由一个左图 L 和一个右图 R 构成,记为 $p := (L, R)$.与字符串文法相似,将一个图中与某个产生式左图同构的子图用此产生式右图进行替换后生成新图的过程称为推导(正向使用产生式规则),反之则为归约(逆向使用产生式规则).图的推导和归约统称为图变换.应用产生式规则对其进行变换的图称为主图.在图变换过程中,主图中与某个产生式左图或右图同构的可替换子图通常称作图柄(redex).从初始图开始,通过正向使用图产生式规则进行推导,可以生成一个图的集合(图语言);反之,对一个图反向使用产生式规则进行归约,可以判断此图是否可由给定图文法来生成.

一般地,上下文相关图产生式是指左右两端均为图(而非单个结点)的产生式,而上下文相关图文法由一组上下文相关图产生式所构成.由于可视化语言句子的空间关系图和抽象语法图都是有向图,用于描述可视化语言的上下文相关图文法产生式的左右两端也均为有向图.以下将给出有向图及有向图同构的形式定义.

定义 1. $G := (N, E, l, s, t)$ 是定义在标号集 Lb 上的一个有向图(以下简称 $G := (N, E)$).其中, N 是结点集,由两个互不相交的非终结点集 N_{NT} 和终结点集 N_T 组成; E 是边的集合; Lb 由结点标号集 Lb_N 和有向边标号集 Lb_E 组成; 函数 $l: N \cup E \rightarrow Lb$ 为结点和边赋予相应的标号; 函数 $s: E \rightarrow N$ 和 $t: E \rightarrow N$ 将边分别映射到其起点和终点.

定义 2. 两个有向图 $G_1 = (N_1, E_1, l_1, s_1, t_1), G_2 = (N_2, E_2, l_2, s_2, t_2)$ 同构,记为 $G_1 \cong G_2$, 如果存在一个双射 $f: G_1 \rightarrow G_2$, 满足:

- $\forall n \in N_1 (l_1(n) = l_2(f(n)))$;
- $\forall e = (n_1, n_2) \in E_1 (f(e) = (f(n_1), f(n_2)) \wedge l_1(e) = l_2(f(e)))$.

图文法与字符串文法不同之处在于:字符串的上下文相关文法只是一种文法形式(框架);而上下文相关图文法可视为一类图文法,它包含多种不同的形式框架,主要有 LGG, RGG, CAGG 和 EGG.一个图文法形式框架的关键特性在于其产生式的形式、嵌入规则的定义、成员判定问题的求解机制以及语法分析算法的设计.为了论述得简洁,下文将侧重介绍 4 个形式框架中与表达能力相关的内容,包括主图形式、产生式形式、图柄定义

和嵌入规则定义.此外,还将在有向图的基础上对个别形式框架进行重新定义以保持叙述上的一致性.除了都可用于描述可视化语言之外,这些图语法框架的共性还在于它们为解决嵌入问题所引入的策略是相似的,尽管在表述形式上有较大差别.

1.1 LGG文法

LGG文法^[8]是最早用于描述可视化语言的上下文相关图语法,其产生式规则是直接定义在有向图上的.

定义 3. 一个LGG产生式 $p:=(L,R)$ 是定义在同一标号集上的一对有向图,其中 L 和 R 含有一个公共(上下文)子图 C ,满足 $E_C:=E_L \cap E_R, N_C:=N_L \cap N_R, \forall e \in E_C \Rightarrow s(e) \in N_C \wedge t(e) \in N_C$;而且 $L \setminus C \neq \emptyset, R$ 为连通图.

一个用于描述过程流图的LGG文法^[8]如图 1 所示,为简洁起见,忽略了边的标号, $B?$ 和 $C?$ 为结点标号的通配符,其中 $B? \in \{\text{begin, fork, if, Stat}\}, C? \in \{\text{end, assign, fork, join, send, receive, if}\}$.注意到,第 1 条产生式通常称作图语法的公理,产生式的左边用 λ 表示,右边为文法的初始图.除公理以外,其他每个产生式的左右两端都包含一个公共子图(灰色结点部分),用于显式地刻画应用产生式时需要匹配的上下文信息.

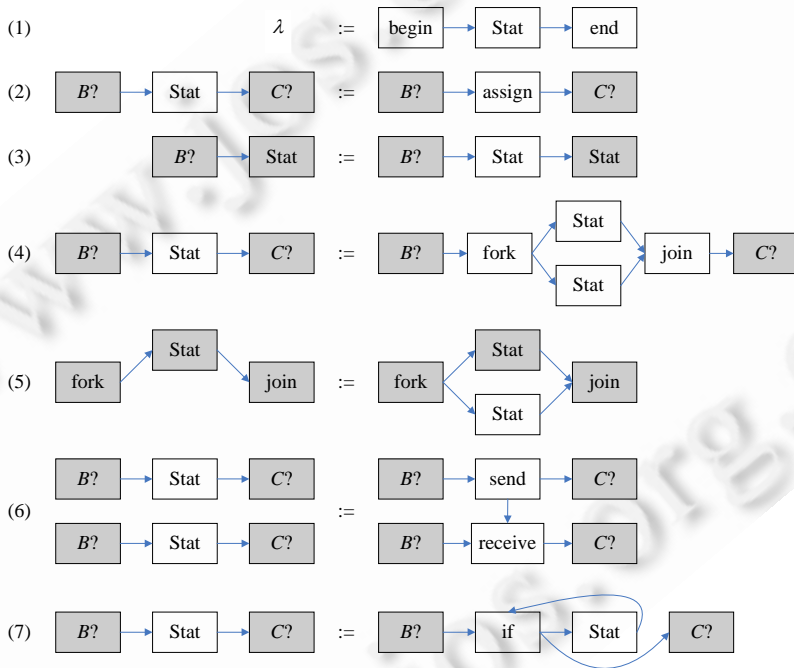


Fig.1 An LGG for specifying process diagram

图 1 一个描述过程流图的 LGG 文法

定义 4. 令 $p:=(L,R)$ 为一个LGG文法产生式,态射 $h:=(h_N, h_E: L \rightarrow G)$ 定义了 L 在 G 中的一个图柄,当且仅当以下条件满足:

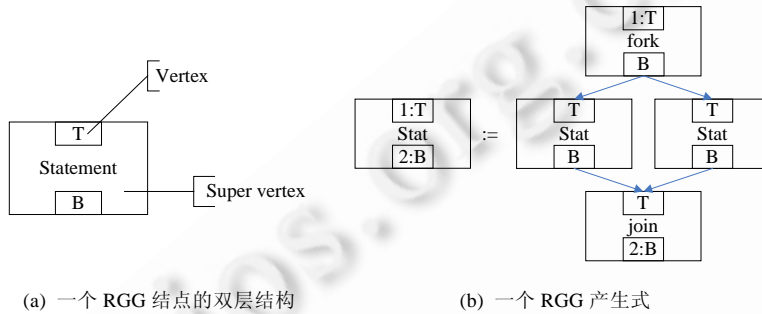
- 悬边条件(dangling edge condition): $\forall n \in N_L \setminus N_R, \forall e \in E_G (s(e) = h_N(n) \vee t(e) = h_N(n) \Rightarrow \exists e' \in N_L \setminus N_R (h_E(e') = e))$;
- 鉴别条件(identification condition): $\forall x \in L \setminus R, \forall x' \in L (h(x) = h(x') \Rightarrow x = x')$.

悬边条件要求,如果主图中有一条边与产生式左(或右)图中某个非上下文结点在图柄中的象相连,则产生式中一定存在一条边与这条边相对应.因此,应用产生式在主图中进行图柄替换时不会产生悬边.

1.2 RGG文法

区别于一般图结点,RGG^[9]引入了一种两层结构结点,如图2(a)所示:标号T和B所对应的内层小方框为顶点,标号Statement所对应的外层方框为超顶点.一个两层结构结点中的顶点和超顶点均可作为连接到该结点的有

向边的连接点;一个结点中的顶点可以有多个,而超顶点只有一个,它表示结点中除顶点以外的连接点.给定一个两层结构结点 n ,其中的顶点 v 记为 $n.v$.由这种结点所构成的两层结构有向图也简记为 G (由于它仅在RGG文法中出现,因此不至于出现混淆现象),并用 $G.V$ 表示 G 中所有结点中的顶点集合.为了解决悬边问题,RGG还在两层结构结点图的基础上引入了一种标记机制和一条嵌入规则.标记机制用于标记产生式结点中的部分顶点或超顶点,顶点 v 的标记记为 $mark(v)$,标记均为正整数,同一图中不同顶点的标记互不相同.一个产生式的左右图中标记相同的顶点标号相同且一一对应,图 1 中产生式(4)的RGG形式如图 2(b)所示.被标记的顶点可以视为上下文元素,在图变换过程中保持不变.



(a) 一个 RGG 结点的双层结构

(b) 一个 RGG 产生式

Fig.2 A node structure and a production of RGG

图 2 RGG 的结点和产生式示例

定义 5. $G:=(N,E,M)$ 是一个标记图,如果 (N,E) 是一个两层结构有向图,而且 $M:G.V \rightarrow I$ 是一个双射.其中, I 是一个有限正整数集合.

定义 6. 一个RGG产生式 $p:=(L,R)$ 是同一标记集上的一对标记图, $L:=(N_L,E_L,M),R:=(N_R,E_R,M)$.

定义 7. 有向图 H 的子图 X 是一个标记图 G 的图柄,如果存在双射 $f:G \rightarrow X$,在 f 下 G 与 X 同构,且

$$\forall v \in G.V.(mark(v)=null) \wedge \forall v_1 \in H.(e=(f(v),v_1) \in H \vee e=(v_1,f(v)) \in H) \Rightarrow e \in X).$$

此外,RGG 文法引入的嵌入规则适用于所有产生式.规则约定,如果与主图中的一个图柄同构的产生式左图或右图中的顶点未标记,则主图中所有连接到此顶点的边必须完全落在图柄中.于是,在进行图变换时,删除(与产生式的一端同构的)图柄的同时保留标记顶点,用产生式的另一端(具有相同的一一对应的标记顶点)替换时覆盖对应标记顶点,从而生成新图.注意到:由于标记顶点被保留,主图中所有连接到标记顶点的边在新图中不会成为悬边;而所有连接到非标记顶点的边都在图柄以内,也不会成为悬边.

1.3 CAGG文法

在有向图的基础上,CAGG^[10]文法引入结点上下文属性来刻画产生式的上下文信息,以解决悬边问题.一般地,可以简单地认为有向图的一个结点上存在两个连结点:入点和出点,入点连接所有的进边,出点连接所有的出边.CAGG文法用上下文属性 ci 来刻画图变换过程中结点中需要保持的连接点,其值域为 $\{up,lp,full,null\}$,分别表示需要保持入点、出点、入点和出点以及空,且规定 $up \cup lp = full$.给定结点 n ,若 $ci(n)$ 不为 $null$,则称 n 为上下文结点.给定边 e , $in(e)$ 和 $out(e)$ 分别表示 e 到达和离开的结点.结点中附带上下文属性的有向图称为属性有向图.

定义 8. 令 G 为一个属性有向图, $CI(G)$ 为图中的上下文结点集合, $Pr(G)=\{P_1, \dots, P_n\}$ 为 G 的一个属性划分,当且仅当 $P_i \in CI(G), P_1 \cup \dots \cup P_n = CI(G), P_j \cap P_k = \emptyset, 1 \leq |P_i| \leq 2, 1 \leq \{i, j, k\} \leq n, j \neq k$,并且如果 P_i 包含两个结点,则它们的 ci 属性分别为 up 和 lp .

定义 9. 一个CAGG产生式 $p:=(L,R,f)$ 是一对属性有向图 L 和 R 以及一个双射 $f.Pr(L)=\{P_1, \dots, P_n\}, Pr(R)=\{Q_1, \dots, Q_n\}$,如果 $f(P_i)=Q_j, 1 \leq \{i, j\} \leq n$,那么,

- $(|P_i|=1 \Rightarrow |Q_j|=1 \vee |Q_j|=2) \wedge |P_i|=2 \Rightarrow |Q_j|=1$;
- $\cup P_i = \cup Q_j$.

定义 10. 一个有向图 H 的子图 S 是一个属性有向图 G 的图柄, 记为 $S \in Rd(H, G)$, 如果存在双射 $f: G \rightarrow S$, 在 f 下 G 与 S 同构, 且满足:

- $\forall n \in G, \forall e \in H((ci(n)=up \vee ci(n)=null) \wedge (out(e)=f(n)) \Rightarrow in(e) \in S)$;
- $\forall n \in G, \forall e \in H((ci(n)=lp \vee ci(n)=null) \wedge (in(e)=f(n)) \Rightarrow out(e) \in S)$.

对于一个 CAGG 产生式而言, L 和 R 属性划分之间的双射 f 建立了左右图中上下文结点上需要保持的连结点之间的一一对应关系. 而且, CAGG 引入了一条与 RGG 类似的适用于所有产生式的嵌入规则. 在进行图变换时, 根据图柄中结点在同构映射 f 下所对应产生式中结点的上下文属性来辨别它的哪些连结点需要保持, 从而在进行子图替换时避免了悬边的出现. 图 1 中产生式(4)的 CAGG 形式如图 3 所示, 其中, 结点的阴影部分对应其 ci 属性值.

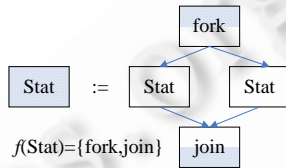


Fig.3 A CAGG production
图 3 一个 CAGG 产生式

1.4 EGG 文法

EGG^[11]文法在有向图的基础上引入了悬边图来构造产生式, 以解决悬边问题. 相关概念定义如下:

定义 11. $G = (N, E, \bar{E})$ 是一个悬边图, 如果 (N, E) 是一个有向图; \bar{E} 是依附于 (N, E) 上的一组悬边的集合, 即 $\forall e \in \bar{E} (in(e) \in N \vee out(e) \in N)$. $G' = (N, E, \bar{E}, M)$ 为标记悬边图, 如果 (N, E, \bar{E}) 是一个悬边图, 双射 $M: \bar{E} \rightarrow I$ 是一个悬边标记函数, 悬边标记集 I 是一个有限正整数集合. 有向图 (N, E) 称为图 G 与 G' 的核图, 记为 $K(G)$.

给定一个有向图 H 及其子图 S , 连接余图 $H \setminus S$ 和子图 S 的 H 中所有边的集合记为 \bar{E} , 由有向图 S 与悬边集 \bar{E} 组成的悬边图称为悬边子图.

定义 12. 一个 EGG 产生式 $p:=(L, R)$ 是同一个悬边标记集上的一对标记悬边图.

定义 13. 有向图 H 的悬边子图 X 是一个悬边图 G 的图柄, 如果存在双射 $f: G \rightarrow X$, 在 f 下 $K(G)$ 与 $K(X)$ 同构, 且 G 与 X 的悬边也在 f 下一一对应.

EGG 也引入了一条类似于 RGG 的嵌入规则. 在进行图变换时, 根据图柄与产生式左图(右图)之间悬边的对应关系以及产生式左右图之间已定义的悬边对应关系, 将右图(左图)替换图柄即可, 这样也可以避免悬边的出现. 图 1 中产生式(4)的 EGG 形式如图 4 所示.

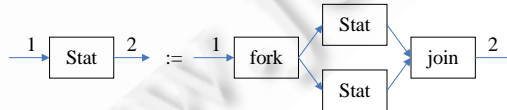


Fig.4 An EGG production
图 4 一个 EGG 产生式

1.5 4种图语法基本特征比较

在一个图语法形式框架中, 与表达能力相关的内容包括主图形式、产生式形式、图柄定义和嵌入规则定义. 下文将通过模拟图生成过程来分析图语法形式框架表达能力之间的关系, 其中涉及到图变换的过程. 综合起来, 与表达能力分析相关的图语法形式框架的基本特征有: 主图的形式、嵌入问题的求解策略、产生式的形式、图柄的形式、图柄匹配、图变换以及嵌入类型. 由于前 4 项特征已在上文中有所论述, 以下将着重分析与比较上述 4 种图语法形式框架在图柄匹配、图变换与嵌入类型上的差异.

从对 4 种图语法形式框架的描述中不难看出,LGG 是通过显式地在左右图上同时附加上下文图元素来定义产生式的,而其他文法则是通过在相关结点上引入顶点、增加属性或悬边来隐式且抽象地刻画产生式的上下文.LGG 通过建立从产生式左(或右)图到主图的态射(而非子图同构)来定义图柄,但同时增加的鉴别条件要求产生式中的非上下文元素(结点或边)与其在主图中的象一一对应.也就是说,将产生式左(或右)图划分为两部分,左右图的非公共部分子图依然与图柄中对应部分同构,公共部分则为态射.因此,在主图中寻找图柄时,不会因为产生式中的不同上下文结点在主图中碰巧出现重叠而遗漏图柄,而其他 3 种图文法则通过构造产生式左(或右)图到两层结构子图、有向子图或悬边子图的同构来定义图柄.而且,在定义图柄时还必须严格地排除可能出现悬边(即所匹配的子图不是图柄)的情形.LGG 设置了悬边条件,RGG 和 CAGG 也定义了相应的约束条件并描述成一条嵌入规则,而 EGG 则要求产生式左(或右)图与图柄的悬边之间一一对应.

在应用产生式进行图变换时,4 种形式的框架所采用的机制是一致的,即用产生式的右图(左图)去替换主图中的图柄(与左图(右图)匹配)以生成新图.由于余图(从主图中删除图柄后所得)中保留了产生式左图(右图)中刻画上下文的信息,包括公共子图、标记顶点、属性对应连接点与悬边,右图(左图)与余图就可以依照上下文信息在产生式左右图之间的对应关系来进行连接.

尽管形式不尽相同,4 种图文法在图柄定义中引入的悬边问题解决机制都可视为一条适用于相应产生式结构形式的不变式嵌入规则^[6].

上述 4 种图文法与表达能力分析相关的基本特征比较结果见表 1.

Table 1 Comparison of basic characteristics for four context-sensitive graph grammars

表 1 4 种上下文相关图语法基本特征比较

	主图形式	嵌入策略(上下文描述机制)	产生式(图对)	图柄形式	图柄匹配	图变换	嵌入规则类型
LGG	有向图	公共子图	包含公共子图的有向图	带公共子图的有向图	公共子图态射+非公共子图同构	公共子图保持	不变式
RGG	两层结构图	两层结点结构+标记机制	标记两层结构图	两层结构图	两层结构子图同构	标记顶点保持	不变式
CAGG	有向图	结点上下文属性	附带结点属性的有向图	有向图	子图同构	上下文属性对应连接点保持	不变式
EGG	有向图	悬边+标记机制	悬边图	悬边图	悬边子图同构	悬边保持	不变式

2 表达能力分析与比较

与字符串文法类似,一个图语法形式框架的表达能力是指它的所有文法实例所描述的图语言的集合.与字符串文法不同,一种图文法的表达能力不仅取决于其产生式的形式,还取决于它的嵌入规则类型.上述 4 种上下文相关图语法形式框架具有相同的嵌入规则,均为不变式,因而其表达能力的差异就仅在于产生式的形式差异.为一般性起见,令 Ω 为所有不变式嵌入类型的上下文相关图语法形式框架的集合.下面首先定义图文法的实例及其所产生的图语言,然后定义表达能力之间的关系.

定义 14. 给定一个图语法形式框架 $GG \in \Omega$, GG 的一个文法实例 $gg := (A, P)$, 其中, A 为文法的初始图, P 为一个符合 GG 形式规范的产生式有限集合. gg 的主图集是由 P 中结点与边所构成的所有图的集合, 记为 H_{gg} .

定义 15. 给定一个图语法形式框架 $GG \in \Omega$ 的一个文法实例 $gg := (A, P)$, gg 所描述的图语言 $L(gg) := \{G | A \mapsto^* G, \forall n \in G(n \in N_T)\}$, 即从 A 出发正向使用图产生式规则生成的所有不含非终结点的图的集合.

与字符串文法不同的是,描述一个可视化语言的图语法可能生成非预期的图^[6,19].也就是说,图语法对于可视化语言的刻画并不总是准确的.基于这一认知,以下分别引入图语言之间的包含和等价关系来定义不同图语法形式框架表达能力之间的关系,包括(严格)不弱于(at least as expressive as)、(严格)强于(more expressive than)、(严格)等价(equally expressive)和不可比较(incomparable)这 4 种情形.其中,严格不弱于、严格强于和严格等价是定义在图语言的等价关系基础之上的,而不弱于、强于、等价和不可比较是定义在图语言的包含关系基础之上的.直观上,前者直接对应于一般意义上的大于等于、大于和等于关系,而后者与前者一一对应,只是更

不严格一些.

鉴于图文法和非单调逻辑之间共有的形式语言特性,文中借鉴了非单调逻辑领域中不同逻辑语言之间表达能力分析的方法^[20-22],通过构造不同图语法形式框架实例之间的转换算法来分析其表达能力之间的关系,而转换算法的时间复杂度必须是多项式级的.

定义 16. 图语言 L_1 与 L_2 等价,记为 $L_1 \approx L_2$,如果 $\forall G \in L_1(\exists G' \in L_2(G \cong G'))$,且 $\forall G \in L_2(\exists G' \in L_1(G \cong G'))$.

定义 17. 图语言 L_1 包含 L_2 ,记为 $L_2 \subseteq L_1$,如果 $\forall G \in L_2(\exists G' \in L_1(G \cong G'))$.

定义 18. 给定上下文相关图语法形式框架 $GG_1, GG_2 \in \Omega$.如果 $\forall gg \in GG_1(\exists gg' \in GG_2(L(gg) \subseteq L(gg')))$,则称 GG_2 的表达能力不弱于 GG_1 ,记为 $GG_1 \rightarrow GG_2$.

定义 19. 给定上下文相关图语法形式框架 $GG_1, GG_2 \in \Omega$.如果 $\forall gg \in GG_1(\exists gg' \in GG_2(L(gg) \approx L(gg')))$,则称 GG_2 的表达能力严格不弱于 GG_1 ,记为 $GG_1 \rightarrow^* GG_2$.

定义 20. 给定图语法形式框架 $GG_1, GG_2 \in \Omega$.如果 $GG_1 \rightarrow GG_2$ 但 $\neg(GG_2 \rightarrow GG_1)$,则称 GG_2 的表达能力强于 GG_1 ,记为 $GG_1 \rhd GG_2$;如果 $GG_1 \rightarrow GG_2$ 且 $GG_2 \rightarrow GG_1$,则称 GG_1 和 GG_2 的表达能力等价,记为 $GG_1 \rightleftharpoons GG_2$;如果 $\neg(GG_1 \rightarrow GG_2)$ 且 $\neg(GG_2 \rightarrow GG_1)$,则称 GG_1 和 GG_2 表达能力不可比较,记为 $GG_1 \not\rightleftharpoons GG_2$.

定义 21. 给定图语法形式框架 $GG_1, GG_2 \in \Omega$.如果 $GG_1 \rightarrow^* GG_2$ 但 $\neg(GG_2 \rightarrow GG_1)$,则称 GG_2 的表达能力严格强于 GG_1 ,记为 $GG_1 \rhd^* GG_2$;如果 $GG_1 \rhd^* GG_2$ 且 $GG_2 \rhd^* GG_1$,则称 GG_1 和 GG_2 的表达能力严格等价,记为 $GG_1 \rightleftharpoons^* GG_2$.

引理 1. 给定图语法形式框架 $GG_1, GG_2 \in \Omega$.如果 $GG_1 \rightarrow^* GG_2$,那么 $GG_1 \rightarrow GG_2$.

引理 2. 给定图语法形式框架 $GG_1, GG_2, GG_3 \in \Omega$.如果 $GG_1 \rightarrow GG_2, GG_2 \rightarrow GG_3$,那么 $GG_1 \rightarrow GG_3$.如果 $GG_1 \rightarrow^* GG_2, GG_2 \rightarrow^* GG_3$,那么 $GG_1 \rightarrow^* GG_3$.

易见,引理 1 和引理 2 的证明可直接从定义 16~定义 19 中获得.

引理 3. 给定图语法形式框架 $GG_1, GG_2 \in \Omega$ 以及图语法实例 $gg_1 := (A_1, P_1) \in GG_1, gg_2 := (A_2, P_2) \in GG_2$.如果

$$\forall G_i \in H_{gg_1} \forall p \in P_1((G_i \mapsto^{p'} G_{i+1}) \Rightarrow \exists G'_i \in H_{gg_2} \exists p' \in P_2(G'_i \cong G_i \wedge G'_i \mapsto^{p'} G'_{i+1} \wedge G'_{i+1} \cong G_{i+1})),$$

其中, H_{gg_1} 和 H_{gg_2} 为对应图语法实例的主图集,那么 $L(gg_1) \subseteq L(gg_2)$.

证明:考虑任一有向图 $G \in L(gg_1)$.由定义 15 可知,一定存在一个从 A_1 出发使用 P_1 中产生式生成 G 的有限推导序列 $A_1 = G_1 \mapsto^{p_1} G_2 \mapsto \dots \mapsto^{p_{n-1}} G_n = G, n \geq 1$.根据前提假设,可令 $G' \in H_{gg_2}$ 且 $G' \cong G$.相应地,可以构造一个基于 gg_2 图语法生成 G' 的推导序列:

(I) $A_2 = G'_1 = G_1$;

(II) 对于原始推导序列中的任一 G_i 以及任一 $p_{ji} \in P_1, 1 \leq i \leq n-1$,如果 $G_i \mapsto^{p_{ji}} G_{i+1}$,根据前提假设, $\exists G'_i \in H_{gg_2}(G'_i \cong G_i)$ 以及 $p'_{ji} \in P_2$,满足 $G'_i \mapsto^{p'_{ji}} G'_{i+1}$,所以 $G'_{i+1} \in H_{gg_2}$.由前提假设可知, $G'_{i+1} \cong G_{i+1}$,因而可以用每一组 G'_i, p'_{ji} 和 G'_{i+1} 分别替换原始推导序列中的对应项.

于是可以得到 $A_2 = G'_1 \mapsto^{p'_{11}} G'_2 \mapsto \dots \mapsto^{p'_{n-1}} G'_n = G'$,因而 $G' \in L(gg_2)$,所以 $L(gg_1) \subseteq L(gg_2)$. \square

注意到,RGG 文法实例所生成的图均为两层结构图(也就是其主图形式),而其他图语法形式框架的实例所生成的均为有向图.给定一个 RGG 文法实例 gg ,易于实现两层结构图 G 与一般有向图之间的转换:删除 G 中所有顶点及其上的标号,就得到与之对应的有向图 G' ;反过来,将 G' 中的结点替换成 gg 中对应标号的结点,就得到两层结构图 G .下文中当出现 RGG 图语法实例时,与其相关的主图和产生式左、右图均为两层结构图.为简洁起见,本文将对此不再进行说明.

2.1 LGG与RGG表达能力比较

分析 LGG 的定义不难看出,产生式中为刻画上下文而引入的公共子图可能不是最简化的.如,一个通配符结点 n 只连接到另外一个或多个通配符结点上,或者两个通配符结点之间存在一条连接边 e .显然,上述结点 n 和边 e 均可被删除而不影响对于产生式上下文的刻画.为简单起见,下面来定义最简化的 LGG 产生式形式.

定义 22. 一个LGG文法实例 $gg:=(A,P)$ 是良好定义的,如果 $\forall p=(L,R)\in P$,令 L 与 R 的公共子图为 C , L 和 R 中的公共子图分别记为 C_L 和 C_R ,那么,

- $\forall n\in C_L(l(n)$ 为通配符), $\exists e\in E_L(s(e)=n\wedge t(e)=n'\vee t(e)=n\wedge s(e)=n')\Rightarrow l(n')$ 不是通配符;
- C_R 中的结点也相应地满足上述约束.

下文中若未作特别说明,均指良好定义的 LGG 文法形式.

此外,与 RGG 图文法形式框架不同,LGG 显式地使用图元素来刻画上下文,且通过构造态射而非同构来定义图柄.为了弥合两者间的差异,定义 LGG 的一个变种 VLGG.

定义 23. VLGG 形式框架是 LGG 的一个变种,差别仅在于,VLGG 将定义 4 中的态射改为双射,同时删除了鉴别条件.

引理 4. 若图文法实例 $gg_1:=(A_1,P_1)\in LGG$,那么一定存在 $gg_2:=(A_2,P_2)\in VLGG$,满足 $L(gg_1)\approx L(gg_2)$.

证明:若图文法实例 $gg_1:=(A_1,P_1)\in LGG$,首先构造一个与之对应的 $gg_2:=(A_2,P_2)$,其中, $A_2=A_1,P_2$ 的构造如下:

对于 $\forall p=(L,R)\in P_1,C$ 为 L,R 的公共子图:

- (1) $p\in P_2$;
- (2) 对于 $\forall n_1,n_2\in N_C(l(n_1)\cap l(n_2)\neq\emptyset)$,同时将 L,R 中的 n_1,n_2 叠加成一个结点,其标号为 $l(n_1)\cap l(n_2)$,由此生成的新产生式 p' 称为 p 的变种, $p'\in P_2$.

令 $(A_2,P_2)\in VLGG$.考虑任意一个图 $G\in L(gg_1)$.由定义 15 可知,一定存在一个从 gg_1 的初始图 A_1 出发使用 P_1 中产生式生成 G 的一个有限推导序列 $A_1 = G_1 \xrightarrow{p_{i_1}} G_2 \xrightarrow{\dots} \xrightarrow{p_{i_{n-1}}} G_n = G, n \geq 1$. 重构一个相应的 (A_2,P_2) 图文法的推导序列如下:(I) $A_2=A_1$;(II) 对于任一 $G_i, 1 \leq i \leq n$,必然存在它的某个子图 $S \subseteq G_i$ 是产生式 p_{j_i} 左图在LGG意义上的图柄.由定义 4 可知,存在一个从 $p_{j_i}.L$ 到 S 的满足鉴别条件的态射 f .若 f 为同构映射,则依然应用产生式 p_{j_i} 进行推导;否则,也就是 f 为非同构的态射.由 P_2 的构造步骤(2)可知,一定存在 p_{j_i} 的一条变种产生式 p'_{j_i} 使得 $p'_{j_i}.L$ 与 S 同构,因而用产生式 p'_{j_i} 替换原序列中的 p_{j_i} .如此,便可得到一个VLGG意义上从 (A_2,P_2) 生成图 G 的推导序列.

反过来,对于任意一个图 $G\in L(gg_2)$,也可类似地构造一个LGG意义上的从 (A_1,P_1) 生成图 G 的推导序列.因此, $L(gg_1)\approx L(gg_2)$. □

尽管 VLGG 文法和 RGG 文法都是通过建立图同构来定义图柄的,但二者在产生式的形式上依然存在很大的差别.为了弥合它们之间的差异,首先定义一种介乎两者之间的文法形式 VRGG(1):RGG 文法的一种变种形式,进而研究 VLGG 和 VRGG(1)在表达能力上的关联;然后探讨 VRGG(1)与 RGG 文法表达能力之间的关系;如此便可以建立起 LGG 和 RGG 表达能力之间的关系.

定义 24. VRGG(1)是 RGG 图文法形式框架的一个变种,不同之处在于它修改了 RGG 结点中顶点的定义:

- 对于 VRGG(1)产生式中任一顶点 $v, l(v)$ 是一个通配符,而不是一般意义上的标号;
- 在图柄匹配时,对于产生式中标记顶点 v ,如果 $l(v)$ 是一个非 ALL?通配符,则主图中与顶点 v 的象(同构映射下)相连的边必须唯一,且与该边相连的另一端结点标号属于 $l(v)$;如果 $l(v)=ALL?$,则主图中与顶点 v 的象相连的边可为 0 或多条.其中,ALL?为通配符,是一个文法实例中所有终结点与非终结点标号集合.

算法 1. VLGG 到 VRGG(1)的转换.

输入:一个VLGG文法实例 $gg_1:=(A_1,P_1)$;

输出:一个相应的VRGG(1)文法实例 $gg_2:=(A_2,P_2)$.

```
{
1   $P_2 := \emptyset$ ;
2  ALL?:= $gg_1$ 中所有终结点与非终结点的标号集;
3  for  $\forall p=(L,R)\in P_1$  {
4    if ( $\forall n\in N_p(l(n)$ 是通配符 $\Rightarrow$ 存在唯一的 $n'$ 与 $n$ 相连且 $l(n')$ 不是通配符)) { //情形 1
5       $i:=1$ ;
```

```

6   for  $\forall n \in N_L \cap N_C$  { //C为L和R的公共子图
7     if ( $l(n)$ 不是通配符) then {
8       在 $p.L$ 中结点 $n$ 上附加 2 个顶点 $v_1, v_2$ ;
9       在 $p.R$ 中结点 $n'$ (即 $f(n)$ )上附加 2 个顶点 $v_1, v_2$ ; //L和R中的公共子图部分之间蕴含了一个
                                     同构映射,记为  $f$ 
10       $d(v_1):=in; l(v_1):=ALL?; mark(v_1):=i; //d(v)$ 指明连接到顶点 $v$ 的边的方向, $in$ 为进边, $out$ 为出边
11       $d(v_2):=out; l(v_2):=ALL?; mark(v_2):=i+1;$ 
12       $i:=i+2;$ 
13      for  $\forall e \in E_L$  {
14        if ( $s(e)=n$ ) then  $s(e):=n.v_1;$ 
15        if ( $t(e)=n$ ) then  $t(e):=n.v_2;$ }
16      for  $\forall e \in E_R$  {
17        if ( $s(e)=n'$ ) then  $s(e):=n'.v_1;$ 
18        if ( $t(e)=n'$ ) then  $t(e):=n'.v_2;$ }
19      else { //l(n)为通配符的情形
20        if  $\exists e \in E_L(s(e)=n) \wedge \exists e' \in E_R(s(e')=n'=f(n))$  then {
21          在 $t(e)$ 上附加一个新顶点 $v_3; d(v_3):=in; t(e).v_3:=l(n); mark(v_3):=i;$ 
22          在 $t(e')$ 上附加一个新顶点 $v_4; d(v_4):=in; t(e).v_4:=l(n); mark(v_4):=i;$ 
23        if  $\exists e \in E_L(t(e)=n) \wedge \exists e' \in E_R(t(e')=n')$  then {
24          在 $s(e)$ 上附加一个新顶点 $v_3; d(v_3):=out; s(e).v_3:=l(n); mark(v_3):=i;$ 
25          在 $s(e')$ 上附加一个新顶点 $v_4; d(v_4):=out; s(e).v_4:=l(n); mark(v_4):=i;$ 
26          从  $p$  中删除结点  $n, n'$  和有向边  $e, e'$ ;
27           $i:=i+1;$ }
28      else { //情形 2,生成包含通配符的产生式
29         $i:=1;$ 
30        for  $\forall n \in N_L \cap N_C$  {
31          同第 8 行~第 18 行;}
32       $P_2:=P_2 \cup \{p\};$ 
33 for  $\forall n \in N_{P_2}$  { //  $N_{P_2}$  为  $P_2$  的结点集
34    $vts(n):=\emptyset;$  //  $vts(n)$  为结点  $n$  的最终顶点集
35   for ( $n^{(i)}$ ) { //  $n^{(i)}$  表示结点  $n$  在  $P_2$  中第  $i$  次出现
36      $vts(n):=vts(n) \cup vt(n^{(i)});$  //  $vt(n^{(i)})$  为  $n^{(i)}$  的顶点集合
37   for  $\forall n \in N_{P_2}$  {
38     for ( $n^{(i)}$ ) {  $vt(n^{(i)}):=vts(n);$  }
39 将  $P_2$  中结点标号中包含通配符的产生式实例化成若干不含通配符的产生式;
}

```

经过简单分析可知,算法 1 的时间复杂度为 $O(mn)$,其中, m 为 LGG 文法实例中产生式的数目, n 为所有产生式左、右图的最大结点数目。

引理 5. 若图文法实例 $gg_1 \in \text{LGG}$,应用算法 1 进行转换后得到文法实例 gg_2 ,则 $gg_2 \in \text{VRGG}(1)$ 。

证明:算法 1 描述了从 VLGG 文法实例到 RGG 文法实例之间的转换过程,包括两个部分,其中,第 1 个 for 循环(第 3 行~第 32 行)实现了从一个 VLGG 产生式到 RGG 产生式的变换,之后的部分(第 33 行~第 38 行)完成了所有产生式结点中顶点的统一布置.给定一个 VLGG 文法实例 gg_1 ,每个产生式 p 的左、右图中的公共子图的结点之间

一一对应.因此,转换之后产生式的形式符合RGG定义.但由于顶点标号被设置成通配符,且顶点标号需要参加图柄匹配,因而转换后得到的文法实例 gg_2 为VRGG(1). \square

引理 6. 图文法形式框架VRGG(1)的表达能力严格不弱于VLGG,即 $VLGG \rightarrow^* VRGG(1)$.

证明:令 $gg_1 := (A_1, P_1) \in VLGG$,对 gg_1 应用算法 1 后生成 $gg_2 := (A_2, P_2)$.由引理 5 可知: P_2 中均为VRGG(1)形式的产生式; $L(gg_2)$ 中均为两层结构的有向图,且边都连接到结点中标号为ALL?的顶点上.

考虑任一有向图 $G_i \in H_{gg_1}$.用 P_2 中的双层结构结点替换 G_i 中的相同标号结点,并将边连接到对应顶点上,可以生成一个与 G_i 同构的两层结构图 $G'_i = tl(G_i)$,令 tl 为从 G_i 到 G'_i 的同构映射函数.显然, $G'_i \in H_{gg_2}$.

假设 $G_i \mapsto^p G_{i+1}$,必然存在某个子图 $S \subseteq G_i$ 是 $p.L$ 在VLGG意义上的图柄,即 $p.L$ 与 S 同构,不妨设 f 为从 $p.L$ 到 S 的同构映射.此外,令 P_2 中与 p 对应的产生式为 p' (由算法 1 可知,若 p 是在情形 1 下被转换的,则转换后得到唯一产生式 p' ;若 p 在情形 2 下被转换,则转换后得到一个包含通配符的产生式,进行实例化后得到一组产生式,根据子图 S 可找到唯一与之对应的产生式 p'),再令 $S' = tl(S) \subseteq G'_i$.接下来证明,用 G'_i 和 p' 替换原始序列中的 p 和 G_i 之后,所生成的图 G'_{i+1} 与 G_{i+1} 同构.

如果 p' 是由 p 在情形 2 下转换得到的,根据算法 1, p 和 p' 的左右子图对应同构,再由LGG与RGG的子图替换过程可知,图 G'_{i+1} 与 G_{i+1} 同构.下面将具体分析 p' 是由 p 在情形 1 下转换得到的情况.

第 1 步,在 S' 中寻找 $p'.L$ 的图柄.构造子图 $S'' \subseteq S'$ 如下:对于 p 中 $\forall n \in N_L$,如果 $l(n)$ 为通配符,则删除 S' 中的结点 $tl(f(n))$ 以及 S' 中与其相连的边.由算法 1 中从 p 生成 p' 的过程可知, $p'.L$ 与 S'' 同构,不妨设同构映射为 f' .而且,考虑 G'_i 与 S'' 之间的连接边, $\forall e \in E_{G'_i} \setminus E_{S''} \setminus E_{G'_i \setminus S''} (e \in E_{S'} \setminus E_{S''} \vee e \in E_{G'_i} \setminus E_{S'})$:(1) 如果 $e \in E_{S'} \setminus E_{S''}$ 且 e 与 $N_{S''}$ 中结点 n_1 相连,不妨假设 $tl(e) = n_1 (s(e) = n_1)$ 的情形可类似证明),则 $p.L$ 中必有边 $e' = f^{-1}(tl^{-1}(e)), tl(e') = n_1' = f^{-1}(tl^{-1}(n_1))$,且 $l(s(e'))$ 为通配符.由算法 1 可知, $p'.L$ 中与 $p.L$ 中 n_1' 对应的结点 n_1'' 上必然存在一个标记顶点 v ,且 $d(v) = in, l(v) = l(s(e'))$;(2) 如果 $e \in E_{G'_i} \setminus E_{S''}$ 且 e 与 $N_{S''}$ 中结点 n_2 相连,不妨仍然假设 $tl(e) = n_2$,则 G_i 中必有 $e'' = tl^{-1}(e)$;此外, $p.L$ 中必有 $n_2' = f^{-1}(tl^{-1}(n_2))$,且 $tl(e'') = f(n_2')$.因此, n_2' 必为上下文结点.否则,由定义 4, e'' 不满足VLGG的悬边条件,亦即 S 不是 $p.L$ 的句柄,与假设相矛盾.而且, $l(n_2')$ 必然不是通配符.否则,由算法 1 可知, $p'.L$ 中不存在与 $p.L$ 中 n_2' 对应的结点,而由 $p'.L$ 与 S'' 同构可知, $p'.L$ 中的结点 $f'^{-1}(n_2) = f'^{-1}(tl(f(n_2)))$,矛盾.所以, n_2' 必为上下文结点,且 $l(n_2')$ 不是通配符.由算法 1 可知, $p'.L$ 中与 n_2' 对应的结点记为 n_2'' ,包含带标记顶点 v ,且 $d(v) = in, l(v) = ALL?$.综合上述两种情形,对于 $G'_i \setminus S''$ 与 S'' 之间的任意一条边, $p'.L$ 中均有一个标记顶点与之对应.因此,由VRGG(1)图柄定义可知, S'' 是 G'_i 中 $p'.L$ 的图柄.

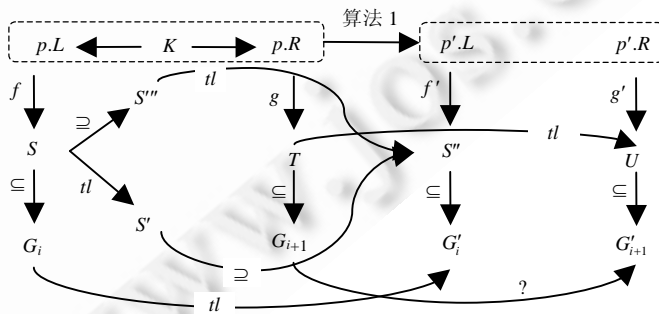


Fig.5 Relationships between the notations in Lemma 6
图 5 引理 6 中符号之间的关系图

第 2 步,证明图 G'_{i+1} 与 G_{i+1} 同构.已知 $G_i \mapsto^p G_{i+1}$,根据LGG的子图替换过程, G_{i+1} 通过用与 $p.R$ 同构的图(不妨设为 T)替换 G_i 中与 $p.L$ 同构的子图 S 而得到.构造 G_{i+1} 中 T 的子图 T' : $\forall n \in N_{p.R}$,如果 $l(n)$ 为通配符,则删除 T 中的结点 $g(n)$ 以及 T 中与其相连的边,其中, g 为从 $p.R$ 到 T 的同构映射;类似地,可以构造 G_i 中 S 的子图 S'' .因而, $G_{i+1} \setminus T' = G_i \setminus S''$.已知 $G'_i \mapsto^{p'} G'_{i+1}$,即用与 $p'.R$ 同构的图,不妨设为 U (令 $g': p'.R \rightarrow U$),替换 G'_i 中的 S'' 得到 G'_{i+1} .因

而, $G'_{i+1} \setminus U = G'_i \setminus S''$. 由引理 5, G'_{i+1} 是一个不含悬边的双层结构有向图. 从算法 1 可知, 在图 $g^{-1}(T') \subseteq p.R$ 中的结点上添加相应的顶点后生成 $p'.R$, 而 $g'(p'.R) = U$, 所以 $U = tl(T')$. 同理得到 $S'' = tl(S')$. 而已知 $G'_i = tl(G_i)$, 故有 $G'_i \setminus S'' = tl(G_i \setminus S')$. 因此, $G'_{i+1} \setminus U = tl(G_{i+1} \setminus T')$.

再考虑连接子图 T' 和 $G_{i+1} \setminus T'$ 之间的边, 由于 $\forall e \in E_{G_{i+1}} \setminus E_{T'} \setminus E_{G_{i+1} \setminus T'} (e \in E_T \setminus E_{T'} \vee e \in E_{G_{i+1}} \setminus E_T)$, 下面分情形进行讨论:

(1) 如果 $e = (s(e), t(e)) = (n_1, n_2) \in E_T \setminus E_{T'}$, 不妨设 $n_1 \in T' (n_2 \in T')$ 的情形也可类似证明), 则 $n_2 \in T$. 由 T' 的定义, $l(g^{-1}(n_2))$ 为通配符. 根据定义 3 (VLGG 产生式定义), 存在唯一边 $e' = (n'_1, n'_2) \in E_{p.L}$ 与 $g^{-1}(e) \in E_{p.R}$ 对应, 其中, $h(n'_2) = g^{-1}(n_2), n'_1$ 与 $g^{-1}(n_1)$ 对应, h 为 p 左、右图的公共子图之间从左到右的同构映射; 相应地, $f(e') \in E_S$, 而且 $f(n'_2) \in N_S \setminus N_{S''}, f(n'_1) \in N_{S''}$. 由算法 1, 存在唯一结点 $n''_1 \in N_{p'.L}$ 与 n'_1 相对应, 且

$$\exists v \in n''_1 (d(v) = out \wedge l(v) = l(g^{-1}(t(e))) \wedge mark(v) \neq null).$$

再令 $n''_1 \in N_{p'.R}$ 为 $p'.R$ 中与 n'_1 对应且具有相同顶点约束的唯一结点. 因为 $S' = tl(S) \subseteq G'_i = tl(G_i), S'' = tl(S')$ 是 G'_i 中 $p'.L$ 的图柄, 所以 $\exists e'' = tl(f(e')) \in S'$, 满足 $t(e'') = tl(f(n'_2)) \in N_{S'} \setminus N_{S''}, s(e'') = f'(n'_1) \in N_{S''}$. 根据 VRGG(1) 的子图替换过程, 用 U 替换 G'_i 中的 S'' 得到 G'_{i+1} 可知, $\exists e''' \in G'_{i+1} (t(e''') = t(e'') \wedge s(e''') = g'(n''_1))$ 与 e'' 唯一对应. 而 $t(e''') = tl(f(h^{-1}(g^{-1}(n_2))))$ 与 n_2 一一对应, n_1 也经由 $g^{-1}(n_1), n'_1, n''_1$ 以及 n''_1 与 $s(e''')$ 唯一对应. 而且易见, 此对应序列可逆. 因此, e 经由 e', e'' 与 e''' 一一对应.

(2) 假设 $e = (n_1, n_2) \in E_{G_{i+1}} \setminus E_T (n_1 \in N_T \wedge n_2 \in N_{G_{i+1}})$ (注意到, $n_1 \in N_{G_{i+1}} \wedge n_2 \in N_T$ 的情形可类似证明), 则

$$g^{-1}(n_1) \in N_K.$$

这里, K 为 p 的左、右图的公共子图.

如果 $l(g^{-1}(n_1))$ 为通配符, 由 VLGG 的图柄定义, $e \in N_{G_{i+1} \setminus T'}$, 而 $G_{i+1} \setminus T' = G_i \setminus S'', tl(G_i \setminus S'') = G'_i \setminus S'' = G'_{i+1} \setminus U$, 故而 $e' = tl(e) \in G'_{i+1} \setminus U$ 与 e 一一对应. 如果 $l(g^{-1}(n_1))$ 不是通配符, 则 $\exists e' = (f(n'_1), n_2) \in E_{G_i} \setminus E_S$ 与 e 一一对应, 其中, $n'_1 = h^{-1}(g^{-1}(n_1)) \in E_{p.L}, f(n'_1) \in N_{S''}, n_2 \in N_{G_{i+1} \setminus T'} = N_{G_i \setminus S'}$. 由算法 1, $\exists n''_1 \in N_{p'.L}$ 与 n'_1 一一对应, 且

$$\exists v \in n''_1 (d(v) = out \wedge l(v) = ALL \wedge mark(v) \neq null).$$

再令 $n''_1 \in N_{p'.R}$ 为 $p'.R$ 中与 n'_1 对应且具有相同顶点约束的唯一结点. 相应地,

$$\exists e'' = tl(e') = (tl(f(n'_1)) = f'(n''_1), tl(n_2)) \in E_{G'_i} \setminus E_S \subseteq E_{G'_i} \setminus E_{S''}.$$

在用 U 替换 G'_i 中 S'' 得到 G'_{i+1} 的过程中, $s(e'') = f'(n''_1) \in N_{S'}$ 被 $g'(n''_1) \in U$ 替代生成 $e''' \in G'_{i+1}$. 因此, e''' 与 e 一一对应.

综上所述, G'_{i+1} 与 G_{i+1} 同构.

所以, $\exists G'_i \in H_{gg_2} (G'_i \cong G_i \wedge G'_i \mapsto^{p'} G'_{i+1} \wedge G'_{i+1} \cong G_{i+1})$. 由引理 3, $L(gg_1) \subseteq L(gg_2)$. 而且易见, $\forall G_i \in H_{gg_2}$, 如果 $G_i \mapsto^{p'} G_{i+1}$, 也可类似证明 $\exists G'_i \in H_{gg_1} (G'_i \cong G_i \wedge G'_i \mapsto^{p'} G'_{i+1} \wedge G'_{i+1} \cong G_{i+1})$, 其中, $p' \in P_1$ 与 $p \in P_2$ 相对应. 因此, 再由引理 3, $L(gg_1) \subseteq L(gg_2)$. 于是根据定义 16, $L(gg_1) \approx L(gg_2)$. 再由定义 19, $VLGG \rightarrow^* VRGG(1)$. □

引理 7. 图文法形式框架 RGG 的表达不弱于 VRGG(1), 即 $VRGG(1) \rightarrow RGG$.

证明: 给定任一图文法实例 $gg_1 := (A_1, P_1) \in VRGG(1)$, 构造一个与之对应的 $gg_2 := (A_2, P_2)$, 如下所示:

- (1) 令 $W_1 = \{w | \exists p \in P_1, \exists n \in N_p, \exists v \in n (l(v) = w \wedge d(v) = in)\}$, 并定义双射 $f: W_1 \rightarrow \{1, \dots, |W_1|\}$; 相应地, 令 $W_2 = \{w | \exists p \in P_1, \exists n \in N_p, \exists v \in n (l(v) = w \wedge d(v) = out)\}$, 并定义双射 $g: W_2 \rightarrow \{1, \dots, |W_2|\}$;
- (2) 对于 $\forall p \in P_1, \forall v \in p$, 如果 $d(v) = in$, 则 $l(v) = in_j$, 其中, $j = f(l(v))$; 如果 $d(v) = out$, 则 $l(v) = out_k$, 其中, $k = g(l(v))$; 修改后的产生式记为 $p', p' \in P_2$.

令 $gg_2 := (A_2, P_2)$, 显然, gg_2 符合 RGG 的文法形式. 不妨设 $gg_2 \in RGG$ (对 gg_2 应用 RGG 的嵌入规则即可). 考虑 $\forall G_i \in H_{gg_1}$, 假设 $G_i \mapsto^{p'} G_{i+1}$, 必然存在它的某个子图 $S \subseteq G_i$ 是 $p.L$ 在 VRGG(1) 意义上的图柄, 用 $p.R$ 替换 S 后生成 G_{i+1} . 不妨假设对 S, G_i 和 p 应用步骤 (2) 后分别生成 S', G'_i 和 p' , 显然, 它们变换前后都对应同构, 而且 $G'_i \in H_{gg_2}$. 事实上, 步骤 (2) 将 VRGG(1) 中通配符顶点标号 (在图柄匹配时, 要求连接到此顶点的边的另一结点标号与该通配符相匹配) 转换成 RGG 意义上的顶点标号 (无需进行上述匹配), 根据定义 24 与 RGG 嵌入规则可知, G'_i 中的子图

S' 必为 $p'.L$ 的图柄,设进行图柄替换后生成图 G'_{i+1} .又因为VRGG(1)与RGG的子图替换过程相同,所以 G'_{i+1} 与 G_{i+1} 同构.

于是由引理 3, $L(gg_1) \subseteq L(gg_2)$.因此,再由定义 18,VRGG(1) \rightarrow RGG. \square

定理 1. 图文法形式框架 RGG 的表达能力不弱于 LGG,即 LGG \rightarrow RGG.

证明:给定任意图文法实例 $gg_1 \in$ LGG.根据引理 4,必定存在 $gg_2 \in$ VLGG满足 $L(gg_1) \approx L(gg_2)$.

由定义 19,LGG \rightarrow^* VLGG.而由引理 6 可知,VLGG \rightarrow^* VRGG(1).因此,由引理 2,LGG \rightarrow^* VRGG(1).再由引理 1,得到LGG \rightarrow VRGG(1).又由引理 7 可知,VRGG(1) \rightarrow RGG.因此,根据引理 2,得到LGG \rightarrow RGG. \square

2.2 RGG与CAGG表达能力比较

上文的分析中已经提到,除了LGG显式地采用公共子图来定义上下文之外,其他3种文法框架均隐式地采用标记顶点、属性或边来描述上下文.通过引入非有向图的方式隐式地刻画产生式的上下文,可以将子图嵌入和替换清晰地隔开,从而使产生式的左、右图准确地刻画了应用产生式时需要匹配和替换的子图.因而,除LGG以外的其余3种文法框架在表达能力方面具有一些共性,不妨记它们及其所有变种形式的集合为 Ω' .

定义 25. 给定一个图文法形式框架 $GG \in \Omega'$ 以及它的一个产生式实例 $p := (L, R)$. p 的基图是从 p 的左、右图中同时删除相应的上下文描述机制,即RGG的顶点标记、CAGG的结点属性或者EGG的悬边之后得到的一对有向图,记为 $p^d := (L^d, R^d)$.

算法 2. CAGG 到 RGG 的转换.

输入:一个CAGG文法实例 $gg_1 := (A_1, P_1)$;

输出:一个相应的RGG文法实例 $gg_2 := (A_2, P_2)$.

```

{
1   $P_2 := \emptyset$ ;
2  for  $\forall p = (L, R, f) \in P_1$  {
3     $i := 1$ ;
4    for  $\forall pr \in Pr(L)$  { //令 $Pr(L) = \{pr_1, \dots, pr_n\}$ ;
5      case  $(|pr|, |f(pr)|)$  {
6        (1,1): 令 $pr = \{n\}$ ;
7          if  $(ci(n) = up)$  then {
8            在 $p.L$ 结点 $n$ 和 $p.R$ 结点 $f(n)$ 上各附加一个顶点 $v_1$ ;  $d(v_1) := in$ ;  $mark(v_1) := i$ ;
9          else if  $(ci(n) = lp)$  then {
10           在 $p.L$ 结点 $n$ 和 $p.R$ 结点 $f(n)$ 上各附加一个顶点 $v_2$ ;  $d(v_2) := out$ ;  $mark(v_2) := i$ ;
11           $i := i + 1$ ;
12        (1,2): 令 $pr = \{n\}$ ,  $f(pr) = \{n_1, n_2\}$ ;
13          在 $p.L$ 结点 $n$ 上附加两个顶点 $v_1, v_2$ ;
14           $d(v_1) := in$ ;  $mark(v_1) := i$ ;  $d(v_2) := out$ ;  $mark(v_2) := i + 1$ ;
15          if  $(ci(n_1) = up)$  then {
16            在 $p.R$ 结点 $n_1$ 上附加一个顶点 $v_1$ ;  $d(v_1) := in$ ;  $mark(v_1) := i$ ;
17            在 $p.R$ 结点 $n_2$ 上附加一个顶点 $v_2$ ;  $d(v_2) := out$ ;  $mark(v_2) := i + 1$ ;
18          else if  $(ci(n_2) = up)$  then {
19            在 $p.R$ 结点 $n_2$ 上附加一个顶点 $v_1$ ;  $d(v_1) := in$ ;  $mark(v_1) := i$ ;
20            在 $p.R$ 结点 $n_1$ 上附加一个顶点 $v_2$ ;  $d(v_2) := out$ ;  $mark(v_2) := i + 1$ ;
21           $i := i + 2$ ;
22        (2,1): 令 $pr = \{n_1, n_2\}$ ,  $f(pr) = \{n\}$ ;
23          if  $(ci(n_1) = up)$  then {

```

```

24      在 $p.L$ 结点 $n_1$ 上附加一个顶点 $v_1$ ;  $d(v_1):=in$ ;  $mark(v_1):=i$ ;
25      在 $p.L$ 结点 $n_2$ 上附加一个顶点 $v_2$ ;  $d(v_2):=out$ ;  $mark(v_2):=i+1$ ;}
26      else if ( $ci(n_2)=up$ ) then {
27          在 $p.L$ 结点 $n_2$ 上附加一个顶点 $v_1$ ;  $d(v_1):=in$ ;  $mark(v_1):=i$ ;
28          在 $p.L$ 结点 $n_1$ 上附加一个顶点 $v_2$ ;  $d(v_2):=out$ ;  $mark(v_2):=i+1$ ;}
29      在 $p.R$ 结点 $n$ 上附加两个顶点 $v_1, v_2$ ;
30       $d(v_1):=in$ ;  $mark(v_1):=i$ ;  $d(v_2):=out$ ;  $mark(v_2):=i+1$ ;
31       $i:=i+2$ ;}
32      删除  $p$  中结点的上下文属性;
33      检查 $p$ 中每个结点是否包含两个顶点 $v_1, v_2$ ;否则,相应地补充,并将边连接到相应的顶点上;
34       $P_2:=P_2 \cup \{p\}$ ;
}

```

由于顶点标号在 RGG 中并无实质性意义,算法 2 中并未对新生成顶点赋予标号.一种简单的处置方法是:对于任一顶点 v ,如果 $d(v)=in$,则 $l(v)=T$;如果 $d(v)=out$,则 $l(v)=B$.易见,算法 2 的时间复杂度为 $O(mn)$,其中, m 为 CAGG 文法实例中产生式的数目, n 为所有产生式左、右图的最大结点数目.

引理 7. 若图语法实例 $gg_1 \in \text{CAGG}$,应用算法 2 进行转换后得到图语法实例 gg_2 ,则 $gg_2 \in \text{RGG}$.

定理 2. 图语法形式框架 RGG 的表达能严格不弱于 CAGG,即 $\text{CAGG} \rightarrow^* \text{RGG}$.

证明:令 $gg_1:=(A_1, P_1) \in \text{CAGG}$,对 gg_1 应用算法 2 后生成 $gg_2:=(A_2, P_2)$.由引理 7, $gg_2 \in \text{RGG}$.考虑任一有向图 $G_i \in H_{gg_1}$,用 P_2 中的双层结构结点替换 G_i 中的对应结点,并将边连接到对应顶点上,可以生成一个与 G_i 同构的两层结构图 $G'_i = tl(G_i)$,令 tl 为从 G_i 到 G'_i 的同构映射.

不妨设 $G_i \mapsto^p G_{i+1}$, $p=(L, R)$,必然存在某个子图 $S \subseteq G_i$ 是产生式左图 L 在 CAGG 意义上的图柄,令 g 为从 L 到 S 的同构映射, $S'=tl(S)$.从算法 2 可知,若产生式 $p'=(L', R') \in P_2$ 从 p 转换得到,则 p^d 与 p'^d 的左、右图对应同构.因此, $S' \subseteq G'_i$ 必与 L' 同构,不妨设 g' 为从 L' 到 S' 的同构映射.由算法 2, $\forall n \in N_p$,如果 $ci(n)=up$,则 p' 的对应结点 n' 中包含一个标记顶点 v_1 , $d(v_1)=in$;如果 $ci(n)=lp(full)$,则 n' 中包含标记顶点 v_2 (v_1 与 v_2).所以, $\forall e \in E_{G'_i} \setminus E_S$.

如果 $t(e)=n \in E_S$,令 $e' = tl^{-1}(e) \in E_{G_i} \setminus E_S$,则 $t(e')=tl^{-1}(n)$.由定义 10,令 $n'=g^{-1}(tl^{-1}(n)) \in N_L$,则 $ci(n)=up \nabla full$.从算法 2 可知,若 L' 中与 n' 对应的结点为 n'' ,则 $n''=g'^{-1}(n)$, n'' 包含顶点 v , $d(v)=in$,且 e 连接到 n 的顶点 v 上.因此, S' 是 L' 的图柄.

由于 $G_i \mapsto^p G_{i+1}$,根据 CAGG 的子图替换过程, G_{i+1} 是通过用与 R 同构的图,设为 X ,替换 G_i 中的 S 得到;再假设 $G'_i \mapsto^{p'} G'_{i+1}$,依据 RGG 子图替换过程, G'_{i+1} 是通过用与 R' 同构的图,设为 Y ,替换 G'_i 中的 S' 得到;令 h, h' 分别为从 R 到 X 、 R' 到 Y 的同构映射,因而 $G_{i+1} \setminus X = G_i \setminus S$, $G'_{i+1} \setminus Y = G'_i \setminus S'$.而 $G'_i \setminus S' = tl(G_i \setminus S)$,所以 $G'_{i+1} \setminus Y = tl(G_{i+1} \setminus X)$.而且由算法 2, $R'^d = tl(R^d)$,所以 $Y = tl(X)$.再考虑连接子图 Y 和 $G'_{i+1} \setminus Y$ 之间的边 $\forall e \in E_{G'_{i+1}} \setminus E_Y \setminus E_{G'_{i+1} \setminus Y}$, $e=(s(e), t(e))=(n_1, n_2)$,不妨设 $n_1 \in N_Y, n_2 \in E_{G'_{i+1} \setminus Y}$ (另一情形可类似证明).那么, $h'^{-1}(n_1) \in N_{R'}$,且 $h'^{-1}(n_1)$ 包含一个顶点 v , $d(v)=out$.由定义 6,令 L' 中与 R' 中 $h'^{-1}(n_1)$ 对应的结点为 n'_1 ,则 n'_1 也包含一个顶点 v' ,满足 $d(v')=out, mark(v')=mark(v)$.再从定义 7 及 RGG 的子图替换过程可得: $\exists e' \in E_{G'_i} \setminus E_S$, $(s(e') = g'(n'_1) \wedge t(e') = n_2)$,因而 $e'' = tl^{-1}(e') \in E_{G_i} \setminus E_S$,且

$$s(e'') = tl^{-1}(g'(n'_1)) \in N_S, t(e'') = tl^{-1}(n_2) \in N_{G_i \setminus S}.$$

又因为 S 是 L 的图柄,根据定义 10, L 中必有一结点 n''_1 ,满足 $g(n''_1) = s(e'')$,且 $ci(n''_1) = lp \nabla full$.再由定义 8 和定义 9,令 $n''_1 \in par_k, 1 \leq k \leq n$,则 $f(par_k) \subseteq N_R$ 中必有唯一结点 n'''_1 与 n''_1 对应,且 $ci(n'''_1) = lp \nabla full$.从 CAGG 的子图替换过程可知, $\exists e''' \in E_{G_{i+1}} \setminus E_X$ 与 e'' 唯一对应,且 $s(e''') = h(n'''_1) \in N_X, t(e''') = tl^{-1}(n_2) \in N_{G_{i+1} \setminus X}$.易见, n_1, n'_1, n''_1 与 n'''_1 之间是一一对应的.因此, e''' 经由 e'', e' 与 e 一一对应.所以, G'_{i+1} 与 G_{i+1} 同构.

于是, $\exists G'_i \in H_{gg_2}$ ($G'_i \cong G_i \wedge G'_i \mapsto^{p'} G'_{i+1} \wedge G'_{i+1} \cong G_{i+1}$).由引理 3, $L(gg_1) \subseteq L(gg_2)$.而且不难看出,上述证明过程是可逆的.因此,类似地有 $L(gg_2) \subseteq L(gg_1)$.根据定义 16, $L(gg_1) \approx L(gg_2)$.再由定义 19, $\text{CAGG} \rightarrow^* \text{RGG}$. \square

2.3 RGG与EGG表达能力比较

定义 26. VRGG(2)是 RGG 图文法形式框架的一个变种,不同之处在于,它对 RGG 中顶点所连接边的数目作了限制:在图柄匹配时,对于产生式中任何标记顶点 v ,主图中与顶点 v 的象(同构映射下)相连的边必须唯一.

算法 3. EGG 到 VRGG(2)的转换.

输入:一个 EGG 文法实例 $gg_1:=(A_1, P_1)$;

输出:一个相应的 VRGG(2)文法实例 $gg_2:=(A_2, P_2)$.

```

{
1   $P_2 := \emptyset$ ;
2  for  $\forall p=(L,R) \in P_1$  { //不妨设  $L=(N_1, E_1, \overline{E_1}, M_1), R=(N_2, E_2, \overline{E_2}, M_2)$ 
3    for  $\forall e \in \overline{E_1}$  { //不妨设  $e' \in \overline{E_2} (M_2(e') = M_1(e))$ 
4      if  $(s(e) \in N_1)$  then {
5        在  $L$  的结点  $s(e)$  上添加一个顶点  $v$ ;  $d(v) := out$ ;
6        在  $R$  的结点  $s(e')$  上添加一个顶点  $v'$ ;  $d(v') := out$ ;}
7      else if  $(t(e) \in N_1)$  then {
8        在  $L$  的结点  $t(e)$  上添加一个顶点  $v$ ;  $d(v) := in$ ;
9        在  $R$  的结点  $t(e')$  上添加一个顶点  $v'$ ;  $d(v') := in$ ;}
10     mark( $v$ ) :=  $M_1(e)$ ; mark( $v'$ ) :=  $M_1(e)$ ;}
11    从  $p$  中删除悬边集  $\overline{E_1}$  与  $\overline{E_2}$ , 以及悬边标记集  $M_1$  与  $M_2$ ;
12     $P_2 := P_2 \cup \{p\}$ ;}
13 for  $\forall n \in N_{P_2}$  { //  $N_{P_2}$  为  $P_2$  的结点集
    //  $c_{in}(n)$  和  $c_{out}(n)$  分别为结点  $n$  中方向为  $in$  和  $out$  的最终顶点数
14     $c_{in}(n) := \max\{c_{in}(n^{(i)})\}$ ; //  $c_{in}(n^{(i)})$  为结点  $n$  在  $P_2$  中第  $i$  次出现时包含的  $in$  的顶点数
15     $c_{out}(n) := \max\{c_{out}(n^{(i)})\}$ ; //  $c_{out}(n^{(i)})$  为结点  $n$  在  $P_2$  中第  $i$  次出现时包含的  $out$  的顶点数
16 for  $\forall n \in N_{P_2}$  {
17   for  $(n^{(i)})$  在  $n^{(i)}$  中添加顶点以使方向为  $in$  和  $out$  的顶点数分别达到  $c_{in}(n)$  和  $c_{out}(n)$ ;}
}

```

不难看出,算法 3 的时间复杂度为 $O(m(n+k))$,其中, m 为 EGG 文法实例中产生式的数目, n 为所有产生式左、右图的最大结点数目, k 为所有产生式中的最大悬边数目.

引理 8. 若图文法实例 $gg_1 \in \text{EGG}$, 应用算法 3 进行转换后得到文法实例 gg_2 , 则 $gg_2 \in \text{RGG}$.

引理 9. 图文法形式框架 VRGG(2) 的表达能力严格不弱于 EGG, 即 $\text{EGG} \rightarrow^* \text{VRGG}(2)$.

证明: 令 $gg_1:=(A_1, P_1) \in \text{EGG}$, 对 gg_1 应用算法 3 后生成 $gg_2:=(A_2, P_2)$. 由定义 26, 不妨令 $gg_2 \in \text{VRGG}(2)$. 考虑任一有向图 $G_i \in H_{gg_1}$, 用 P_2 中的双层结构结点替换 G_i 中的对应结点, 并将边连接到对应顶点上, 可以生成一个与 G_i 同构的两层结构图 $G'_i = tl(G_i)$, 令 tl 为从 G_i 到 G'_i 的同构映射.

不妨假设 $G_i \mapsto^p G_{i+1}$, 其中, $p=(L,R), L=(N_1, E_1, \overline{E_1}, M_1), R=(N_2, E_2, \overline{E_2}, M_2)$, 必然存在某个悬边子图 $S \subseteq G_i$ 是产生式 p 的左图 L 在 EGG 意义上的图柄. 不妨设 f 为从 L 到 S 的同构映射, $K(S)$ 为 S 的核图, 则 $S' = tl(K(S))$; 记 S 的悬边集为 \overline{E} . 从算法 3 可知, 若从 p 转换得到 $p'=(L', R') \in P_2$, 则 L^d 与 L'^d 以及 R^d 与 R'^d 对应同构. 因此, $S' \subseteq G'_i$ 必与 L' 同构, 不妨设 f' 为从 L' 到 S' 的同构映射. $\forall e \in E_{G'_i} \setminus E_{S'}$, 不妨设 $t(e) = n \in E_{S'}$. ($s(e) \in E_{S'}$ 的情形可类似证明), 则 $e' = tl^{-1}(e) \in \overline{E}$. 令 $n' = t(e')$, 必然有 $n' = tl^{-1}(n) \in N_{K(S)}$. 由算法 3, 设与 L 中结点 $f^{-1}(e')$ 对应的 L' 中结点为 n'' , 则 $n'' = f'^{-1}(n)$, 且 n'' 中必有一顶点 v 满足 $d(v) = in$ 与边 e 一一对应. 因此, S' 是 L' 的图柄.

由于 $G_i \mapsto^p G_{i+1}$, 根据 EGG 的子图替换过程, G_{i+1} 是通过用与 R 同构的悬边有向图, 设为 X , 替换 G_i 中的 S 得到的, 令 X 悬边集为 $\overline{E'}$. 再假设 $G'_i \mapsto^{p'} G'_{i+1}$, 依据 RGG 子图替换过程, G'_{i+1} 是通过用与 R' 同构的图, 设为 Y , 替换 G'_i 中的 S'

得到;令从 R 到 X 、 R' 到 Y 的同构映射分别为 g 和 g' ,因而 $G_{i+1} \setminus X = G_i \setminus S, G'_{i+1} \setminus Y = G'_i \setminus S'$.而且由算法 3, $R'^d = tl(R^d)$, 因而 $Y = tl(K(X))$.又由定义 11, $G'_i \setminus S' = tl(G_i \setminus S)$,所以 $G'_{i+1} \setminus Y = tl(G_{i+1} \setminus X)$.考虑连接子图 Y 和 $G'_{i+1} \setminus Y$ 之间的边, $\forall e = (s(e), t(e)) = (n_1, n_2) \in E_{G'_{i+1}} \setminus E_Y \setminus E_{G'_{i+1} \setminus Y} (n_1 \in N_Y \wedge n_2 \in N_{G'_{i+1} \setminus Y})$ (另一情形与此类似),则 $g'^{-1}(n_1) \in N_{R'}$,且 $g'^{-1}(n_1)$ 包含一个顶点 $v, d(v) = out$.由定义 6, L' 中与 R' 中 $g'^{-1}(n_1)$ 对应的结点 n'_1 也包含一个顶点 $v', d(v') = out, mark(v') = mark(v)$.然后,从定义 7 得到 $\exists e' \in E_{G'_i} \setminus E_{S'} (s(e') = f'(n'_1) \wedge t(e') = n_2)$.所以有 $tl^{-1}(e') \in E_{G_i} \setminus E_{K(S)}$,而且 $s(tl^{-1}(e')) = tl^{-1}(f'(n'_1)) \in N_{K(S)}, t(tl^{-1}(e')) = tl^{-1}(n_2) \in N_{G_i \setminus S}$.根据定义 13, $\exists e'' \in \overline{E}_1$ 与 $tl^{-1}(e')$ 唯一对应.令 $s(e'') = n''_1$,则 $n''_1 = f^{-1}(tl^{-1}(f'(n'_1)))$.由算法 3, $M_1(e'') = mark(v') = mark(v)$.再由定义 11 可知, R 中必定存在与 L 中 e'' 对应的边,设为 e''' ,则 $M_2(e''') = M_1(e'') = mark(v)$.令 $s(e''') = n''_2 \in N_2$,因此 $g(e''') \in \overline{E}$,且 $s(g(e''')) = g(n''_2)$;再由EGG子图替换过程得到 $t(g(e''')) = tl^{-1}(n_2)$.注意到,上述结点和边(结点对)均具唯一性,故而 $g(e''')$ 与 e 一一对应.

所以, G'_{i+1} 与 G_{i+1} 同构.归纳起来, $\exists G'_i \in H_{gg_2} (G'_i \cong G_i \wedge G'_i \mapsto^{p'} G'_{i+1} \wedge G'_{i+1} \cong G_{i+1})$.由引理 3, $L(gg_1) \subseteq L(gg_2)$.此外,不难看出,上述证明过程是可逆的.因此,对应地, $L(gg_2) \subseteq L(gg_1)$.再由定义 16, $L(gg_1) \approx L(gg_2)$.所以根据定义 19, $EGG \rightarrow^* VRGG(2)$. □

引理 10. 图文法形式框架 RGG 的表达不弱于 VRGG(2),即 $VRGG(2) \rightarrow RGG$.

证明:令 $gg_1 := (A_1, P_1) \in VRGG(2)$,再令 $gg_2 := (A_1, P_1) \in RGG$,显然, $\forall G_i \in H_{gg_1}$.如果 $G_i \mapsto^p G_{i+1}$,必然有

$$\exists G'_i \in H_{gg_2} (G'_i \cong G_i \wedge G'_i \mapsto^{p'} G'_{i+1} \wedge G'_{i+1} \cong G_{i+1}).$$

因为由定义 26,只需令 $G'_i = G_i, G'_{i+1} = G_{i+1}$ 即可.

根据引理 3, $L(gg_1) \subseteq L(gg_2)$.因此,再由定义 18, $VRGG(2) \rightarrow RGG$. □

例 1:一个流程图语言 L_{CF} :由一对fork和join构成包含任意数目(≥ 2)Stat分支流程图的集合,如图 6 所示.

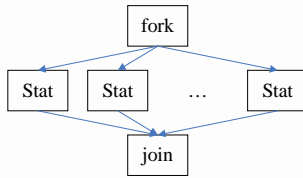


Fig.6 Instances of graph language L_{CF}

图 6 图语言 L_{CF} 的图实例

引理 11. EGG图文法形式框架不能用于描述图语言 L_{CF} .

证明:由定义 14 可知,一个图文法实例所包含的产生式的数量是有限的,不妨设最大值为 m .在构造EGG文法生成图语言 L_{CF} 时,若要生成包含 $i+1 (i \geq 2)$ 个Stat分支的流程图,不妨假设每次使用产生式时新增一个分支(需要特别指出的是,新增若干分支的情况与此类似),则在产生式的上下文结点fork和join上必然已经连接了 i 个Stat结点,即 i 或者 $i-1$ 条(其中一个Stat结点也作为上下文)悬边,其他情形可简化成这两种情形.然而,Stat分支数为 $2 \sim m+2$ 个的流程图均包含在 L_{CF} 中,因而所需要的产生式数量必不少于 $m+1$ (第 1 个产生式生成 2 个分支,然后每个产生式新增一个分支),与产生式数量最大为 m 相矛盾.所以,EGG不能用于刻画流程图语言 L_{CF} . □

定理 3. 图文法形式框架 RGG 的表达能力强于 EGG,即 $EGG \not\rightarrow RGG$.

证明:由引理 9 可知, $EGG \rightarrow^* VRGG(2)$.所以根据引理 1, $EGG \rightarrow VRGG(2)$.而由引理 10 可知, $VRGG(2) \rightarrow RGG$.再根据引理 2, $EGG \rightarrow RGG$.此外, RGG 可用于描述图语言 L_{CF} ,结合引理 11,根据定义 20, $EGG \not\rightarrow RGG$. □

2.4 LGG与EGG表达能力比较

LGG 和 EGG 图文法形式框架在结构上具有较为明显的相似性, LGG 用结点来刻画产生式上下文,而 EGG 则采用悬边的形式来描述上下文;由于不同结点所连接的边有可能相同, LGG 对于上下文的刻画更为精细,因而表达能力应不弱于 EGG.下面将论证这一分析.

算法 4. EGG 到 LGG 的转换.

输入:一个EGG文法实例 $gg_1:=(A_1,P_1)$;

输出:一个相应的LGG文法实例 $gg_2:=(A_2,P_2)$.

```

{
1   $P_2:=\emptyset$ ;
2  for  $\forall p=(L,R)\in P_1$  { //不妨设  $L=(N_1,E_1,\overline{E_1},M_1),R=(N_2,E_2,\overline{E_2},M_2)$ 
3    for  $\forall e\in \overline{E_1}$  { //不妨设  $e'\in \overline{E_2}(M_2(e')=M_1(e))$ 
4      if  $(s(e)\in N_1)$  then {
5        在  $L$  中添加一个新结点  $n; t(e):=n; l(n):=ALL?$ ;
6        在  $R$  中添加一个新结点  $n'; t(e):=n; l(n'):=ALL?;$ 
7      else if  $(t(e)\in N_1)$  then {
8        在  $L$  中添加一个新结点  $n; s(e):=n; l(n):=ALL?;$ 
9        在  $R$  中添加一个新结点  $n'; s(e):=n; l(n'):=ALL?;$ 
10      $N_1:=N_1\cup\{n\}; N_2:=N_2\cup\{n'\};$ 
11     从  $p$  中删除悬边集  $\overline{E_1}$  与  $\overline{E_2}$ , 以及悬边标记集  $M_1$  与  $M_2$ ;
12      $P_2:=P_2\cup\{p\};$ 
}
}

```

易见,算法 4 的时间复杂度为 $O(mn)$,其中, m 为 EGG 文法实例中产生式数目, n 为所有产生式左、右图的最大结点数目.

引理 12. 若图文法实例 $gg_1\in EGG$,应用算法 4 进行转换后得到文法实例 gg_2 ,则 $gg_2\in LGG$.

引理 13. 图文法形式框架LGG的表达能力严格不弱于EGG,即 $EGG\rightarrow^*LGG$.

证明:令 $gg_1:=(A_1,P_1)\in EGG$,对 gg_1 应用算法 4 后生成 $gg_2:=(A_2,P_2)$.由引理 12, $gg_2\in LGG$.

考虑任一有向图 $G_i \in H_{gg_1}$,不妨假设 $G_i \mapsto^p G_{i+1}$,其中, $p=(L,R), L=(N_1,E_1,\overline{E_1},M_1),R=(N_2,E_2,\overline{E_2},M_2)$,必然存在某个悬边子图 $S=(N_3,E_3,\overline{E_3})\subseteq G_i$ 是产生式 p 的左图 L 在 EGG 意义上的图柄,令 f 为从 L 到 S 的同构映射.构造 $S'=(N'_3,E'_3)$ 如下:对于 $\forall e\in \overline{E_3}$,如果 $s(e)\notin N_3$,则 $N'_3=N_3\cup\{s(e)\}$;否则,如果 $t(e)\notin N_3$,则 $N'_3=N_3\cup\{t(e)\}$;再令 $E'_3=E_3\cup\{e\}$,于是 $S'\subseteq G_i$.从算法 4 可知,将 p 转换后得到 $p'=(L',R')\in P_2$,构造从 L' 到 S' 的态射 f' 如下:

- (1) $\forall x\in L, x$ 为边或结点, $f'(x)=f(x)\in S\subseteq S'$;
- (2) $\forall e_1, e_2 \in \overline{E_1}$, 如果 $s(e_1), s(e_2)\in N_1 \wedge t(f(e_1))=t(f(e_2))\in N_3$, 那么 $f'(t(e_1))=f'(t(e_2))\in N'_3$; 如果 $s(e_1), t(e_2)\in N_3 \wedge t(f(e_1))=s(f(e_2))$, 那么 $f'(t(e_1))=f'(s(e_2))$; 如果 $t(e_1), t(e_2)\in N_3 \wedge s(f(e_1))=s(f(e_2))$, 那么 $f'(s(e_1))=f'(s(e_2))$.

其中,步骤(1)与 LGG 图柄定义中的鉴别条件一致,步骤(2)刻画了从 L' 的上下文结点(从算法 4 可知,所有上下文结点的标号均为 ALL?,可与任意标号的结点匹配)到 S' 的态射,所以,根据 LGG 图柄定义, S' 为 L' 的图柄.

由于 $G_i \mapsto^p G_{i+1}$,根据 EGG 的子图替换过程, G_{i+1} 是通过用与 R 同构的悬边有向图,设为 X ,替换 G_i 中的 S 得到的,令从 R 到 X 的同构映射为 g ;假设 $G'_i \mapsto^{p'} G'_{i+1}$,依据 LGG 的子图替换过程, G'_{i+1} 是用某个有向图 Y ,替换 G_i 中 S' 得到的,其中, Y 的构造如下: $Y=(S'\setminus X)\cup X$,且 $S'\setminus X$ 与悬边图 X 之间的连接关系为 $\forall e\in N_T(g^{-1}(e)\in \overline{E_2})$, 如果 $s(g^{-1}(e))\in N_2$, 那么 $t(e)=t(f(e'))\in N'_3$, 其中, $e'\in N_1 (M_1(e')=M_2(g^{-1}(e)))$; 如果 $t(g^{-1}(e))\in N_2$, 那么 $s(e)=s(f(e'))$.相应地,从 R' 到 Y 的态射 g' 构造如下:由算法 4,令 R' 中与 R 对应的悬边子图为 T ,并令从 R 到 T 的同构映射为 h .对于 $\forall x\in T, x$ 为边或结点, $g'(x)=g(h^{-1}(x))\in X$; 对于 $\forall x\in R'\setminus T, x$ 为结点, $g'(x)=f'(x)\in S'$, 其中, $x'\in L'$ 为与 x 对应的通配符结点.从 EGG 和 LGG 的子图替换过程可知, G'_{i+1} 与 G_{i+1} 同构.

于是由引理 3, $L(gg_1)\subseteq L(gg_2)$.类似地,可证 $L(gg_2)\subseteq L(gg_1)$.

根据定义 16, $L(gg_1)\approx L(gg_2)$.因此,由定义 19, $EGG\rightarrow^*LGG$. □

定理 4. 图文法形式框架LGG的表达能力严格强于EGG,即 $EGG\rightleftarrows^*LGG$.

证明:由图 1 可知,LGG可用于描述图语言 L_{CF} ;再由引理 11 可知, $\neg(LGG \rightarrow EGG)$;又由引理 13 可知, $EGG \rightarrow^k LGG$.于是,根据定义 20,得到 $EGG \not\rightarrow^k LGG$. \square

2.5 表达能力综合分析

综合上述分析结果,可以画出 4 种上下文相关图语法表达能力之间的关系图,如图 7 所示.上下文相关图语法表达能力之间的关系以及相应的转换算法可直接服务于图文法的应用.一方面,从语法分析算法上看,LGG的算法极为复杂,时间复杂度呈指数级^[8];引入关键对分析技术(critical pair analysis)固然可以提高算法分析效率,但依然无助于降低时间复杂度^[23].然而在合流条件^[9,10]下,RGG和CAGG语法分析算法(无需回溯)的时间复杂度仅为多项式级;当然,存在不满足合流条件的图语法实例^[23,24],但对RGG和CAGG而言,解决这一问题的可行途径是通过引入语义约束将非合流的产生式集合转换成合流的图语法实例^[24].但合流条件对于EGG而言,限制过强^[11],通常它只能使用带回溯的语法分析算法,因而算法时间复杂度必然大于RGG.另一方面,在产生式的形式上,LGG最为直观但不够简洁;RGG很不直观但较为简洁;EGG和CAGG均比LGG和RGG更为简洁,但又都不及LGG直观.因此从产生式设计角度看,选择LGG,CAGG和EGG较为合适;而从语法分析的角度看,则选择RGG和CAGG更为高效.而上文已经建立了上下文相关图语法形式框架表达能力之间的 4 种关系:LGG严格强于EGG、RGG强于EGG、RGG不弱于LGG以及RGG严格不弱于CAGG;与此同时,还构造了从EGG到LGG、LGG到RGG、EGG到RGG以及CAGG到RGG的转换算法,从而提供了对应图语法实例之间的转换途径.用户在应用图语法时可以自由地选择LGG,CAGG,EGG甚至RGG形式框架来定义图语言;然后,应用转换算法将所设计的非RGG图语法实例转换成对应的RGG实例.如此一来,便可以应用RGG的高效算法实现图的语法分析.由此可见,文中建立的图语法表达能力之间的关系为这一应用模式奠定了必要的理论基础,而相应的转换算法则为它提供了一条切实可行的实现途径.

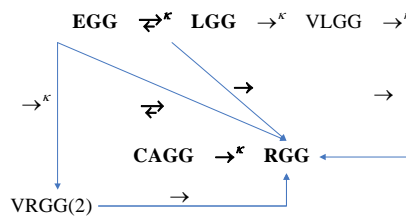


Fig.7 Expressiveness relationships between four graph grammar formalisms

图 7 4 种图语法表达能力之间的关系

3 相关工作

在可视化语言描述方面,早期具有代表性的研究工作有RG文法^[2]、PLG文法^[3]和CMG文法^[4].RG文法在形式文法的基础上加入属性符号集和关系符号集来描述终结符或非终结符的特征,以及它们之间的位置与连接关系.而PLG则基于属性化多重集文法,其产生式左端为一个非终结符,右端是一个符号多重集,且允许出现一个上下文符号;它引入以右端元素属性为参数的语义函数来计算左端元素的属性,并利用约束来刻画右端元素属性之间的关系.CMG文法产生式的形式与PLG类似,不同之处在于它没有语义函数,但在约束中引入了作用于符号的存在量词来刻画应用产生式时所必须的上下文.其中,RG文法为上下文无关文法,PLG和CMG为上下文相关文法.与上下文相关图语法相比,最大的差别在于它们是用一维的字符串而非图的形式来刻画二维的图;而且,本质上,它们刻画的是空间关系图而非抽象语法图;而且在表达能力上,它们甚至不能正确地描述过程流图^[8].

从图语法理论研究的角度来看,嵌入问题是设计图语法形式面临的关键问题之一,而嵌入问题的一般解决途径是引入嵌入规则.图文法的表达能力既取决于图语法形式又取决于嵌入规则类型.Nagl^[25]已经证明了配备不同嵌入规则的不同形式图语法所生成的图语言之间的关系,亦即不同图语法表达能力之间的关系.比如,若限

定嵌入规则为无限制型,则形式无限制型图文法的图语言为递归可枚举图集,上下文无关图文法与上下文相关图文法的图语言相同,均真包含于递归可枚举图集;而且在不变型嵌入规则下,上下文无关图文法的图语言真包含于上下文相关图文法。然而,若产生式形式为无限制型,则配备任何嵌入规则所生成的图语言都相同,均为递归可枚举图集^[26]。从研究内容上看,上述研究工作建立了图文法表达能力之间关系的体系,即不同图文法形式配备不同嵌入规则所生成图语言之间的关系;在这一表达能力体系中,上下文相关文法被粗粒度地视为一种文法形式(框架),而本文的工作建立了上下文相关图文法不同形式框架表达能力之间的关系,其中,上下文相关图文法被细粒度地视为一类文法,它包含多种不同的形式框架,每种框架可视为上下文相关图文法的一种表现形式,对应的嵌入规则类型为不变式。因此,本文的工作是对Nagl等人理论工作的进一步精化和补充。从研究方法上看,二者之间既具有一定的相似性,也存在较大的差异。相似之处在于,二者都用一种图文法形式框架去模拟另一种形式框架的图生成过程来比较两种形式框架表达能力之间的关系;而不同之处则源于研究内容上的差异。上述相关工作中涉及的图文法产生式形式都是一般意义上的有向图^[25,26],但嵌入规则都较为复杂,因而表达能力比较的关键在于嵌入规则的操纵能力的构造。大多采用的研究方法是:对于一种图文法类型的产生式,根据它的嵌入规则类型设计适当的嵌入表达式来模拟另一种图文法类型的产生式,进而构造性地证明二者在图生成能力上的关系。而本文研究所涉及的上下文相关图文法形式框架在产生式的形式上差异较大,大部分都不是一般意义上的有向图,而嵌入规则均为不变式,因而表达能力比较的关键在于不同产生式形式之间的转换。为此,我们借鉴了非单调逻辑中不同形式系统之间表达能力的分析方法,通过构造不同图文法形式框架文法实例之间的转换算法,用一种形式框架的文法实例来模拟另一种文法实例的图生成过程,从而构造性地证明二者表达能力之间的关系。

还有一些研究工作关注于对已有上下文相关图文法形式框架的扩充。Bottoni提出了扩充的LGG图文法CLGG(contextual LGG)^[23]。在LGG的基础上,CLGG引入了否定应用条件(negative application condition)、复杂谓词(complex predicate)和集合结点(set node)等产生式应用条件,以实现更为简洁地描述可视化语言,但这也导致图柄匹配过程更为复杂以及产生式难以逆向使用的问题^[6]。而且,新引入的应用条件和嵌入规则是图重写中常见的,也同样适用于其他上下文相关图文法形式框架。此外,在RGG图文法的形式框架上,Kong提出了SGG文法^[18](spatial graph grammar),它通过引入拓扑关系、方位关系、距离关系以及排列关系来进一步刻画产生式中结点之间的空间关系。由于添加了空间描述机制,SGG比RGG表达能力更强,它不仅能表示抽象语法图,还能表示空间关系图。需要指出的是,这一空间描述机制不仅适用于RGG,也适用于其他形式框架,从而在必要时扩充它们的表达能力。因而,文中对于表达能力的讨论只局限于基本的采用不变型嵌入规则的图文法形式框架。

4 结 论

本文在对已有上下文相关图文法形式框架的基本特征进行分析和归纳的基础上,通过构造图文法之间的转换算法,证明了这些形式框架表达能力之间的关系。从理论研究的角度看,这一结果是对已有上下文相关图文法理论的补充;而从应用的角度看,因研究表达能力而构造的算法更进一步为不同图文法实例之间进行转换提供了一条切实可操作的途径。这一转换途径让原本隔离的不同图文法之间可以相互关联起来,使图文法的应用不必再拘泥于一个形式框架,而是根据应用需求选择不同的形式框架进行应用描述和分析,从而提高图文法的易用性和分析效率。

表达能力的分析一直是形式理论中较为复杂的一个问题。尽管文中的结论已在很大程度上揭示了上下文相关图文法表达能力之间的关系,但仍有两个问题值得更进一步的理论探究:(1) CAGG与EGG以及LGG表达能力之间的关系。由于它们在上下文的描述形式及其含义上都存在很大差异,直接通过形式转换的方法进行比较是非常困难的;(2) 表达能力之间的关系中是否还存在其他的真包含关系。真包含关系的建立需要构造一个无法描述的反例,而这在可以描述无限上下文图元的图文法形式中是难以构造的。

致谢 感谢南京大学马晓星教授、河海大学曾晓勤教授对本文工作的贡献。感谢审稿专家的建设性意见。

References:

- [1] Chang SK. Visual languages: A tutorial and survey. *IEEE Software*, 1987,4(1):29–39. [doi: 10.1109/MS.1987.229792]
- [2] Crimi C, Guercio A, Nota G, Pacini G, Tortora G, Tucci M. Relation grammars for modelling multi-dimensional structures. In: *Proc. of the '90 IEEE Workshop on Visual Languages*. Los Alamitos: IEEE Computer Society Press, 1990. 168–173. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=128400> [doi: 10.1109/WVL.1990.128400]
- [3] Golin EJ, Reiss SP. The specification of visual language syntax. In: *Proc. of the '89 IEEE Workshop on Visual Languages*. Washington: IEEE Computer Society Press, 1989. 105–110. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=77050> [doi: 10.1016/S1045-926X(05)80013-8]
- [4] Marriott K. Constraint multiset grammars. In: *Proc. of the '94 IEEE Symp. on Visual Languages*. IEEE Computer Society Press, 1994. 118–125. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=363633> [doi: 10.1109/VL.1994.363633]
- [5] Rekers J, Schürr A. A graph based framework for the implementation of visual environments. In: *Proc. of the '96 IEEE Symp. on Visual Languages*. Los Alamitos: IEEE Computer Society Press, 1996. 148–155. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=545281> [doi: 10.1109/VL.1996.545281]
- [6] Blostein D, Fahmy H, Grbavec A. Practical use of graph rewriting. Technical Report, 1995. 95–373.
- [7] Ehrig H, Engels G, Kreowski HJ, Rozenberg G. *Handbook of Graph Grammars and Computing by Graph Transformation, Vol.2: Applications, Languages and Tools*. River Edge: World Scientific, 1999.
- [8] Rekers J, Schürr A. Defining and parsing visual languages with layered graph grammars. *Journal of Visual Languages and Computing*, 1997,8(1):27–55. [doi: 10.1006/jvlc.1996.0027]
- [9] 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.
- [10] Zou Y, Zeng XQ, Han XQ, Zhang K. Context-Attributed graph grammar framework for specifying visual languages. *Journal of Southeast University (English Edition)*, 2008,24(4):455–461.
- [11] Zeng XQ, Han XQ, Zou Y. An edge-based context-sensitive graph grammar formalism. *Journal of Software*, 2008,19(8): 1893–1901 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/19/1893.htm> [doi: 10.3724/SP.J.1001.2008.01893]
- [12] Zhao CY, Kong J, Zhang K. Program behavior discovery and verification: A graph grammar approach. *IEEE Trans. on Software Engineering*, 2010,36(3):431–448. [doi: 10.1109/TSE.2010.3]
- [13] Kong J, Zhao CY. Visual language techniques for software development. *Journal of Software*, 2008,19(8):1902–1919 (in English with Chinese abstract). <http://www.jos.org.cn/1000-9825/19/1902.htm> [doi: 10.3724/SP.J.1001.2008.01902]
- [14] Song GL, Zhang K, Kong J. Model management through graph transformations. In: Bottoni P, Hundhausen C, Levialdi S, Tortora G, eds. *Proc. of the 2004 IEEE Int'l Symp. on Visual Languages and Human-Centric Computing*. Washington: IEEE Computer Society Press, 2004. 75–82. [doi: 10.1109/VLHCC.2004.37]
- [15] Zhao CY, Kong J, Dong J, Zhang K. Pattern based design evolution using graph transformation. *Journal of Visual Languages and Computing*, 2007,18(4):378–398. [doi: 10.1016/j.jvlc.2007.07.004]
- [16] Zhang K, Zhang DQ, Deng Y. A visual approach to XML document design and transformation. In: *Proc. of the 2001 IEEE Int'l Symp. on Human-Centric Computing Languages and Environments*. Washington: IEEE Computer Society Press, 2001. 312–319. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=995279> [doi: 10.1109/HCC.2001.995279]
- [17] Zhang K, Kong J, Qiu MK, Song GL. Multimedia layout adaptation through grammatical specifications. *Multimedia Systems*, 2005, 10(3):245–260. [doi: 10.1007/s00530-004-0155-2]
- [18] Kong J, Zhang K, Zeng X. Spatial graph grammars for graphical user interfaces. *ACM Trans. on Computer-Human Interaction*, 2006,13(2):268–307. [doi: 10.1145/1165734.1165739]
- [19] Rekers J, Schürr A. A graph grammar approach to graphical parsing. In: Haarslev V, ed. *Proc. of the 11th IEEE Int'l Symp. on Visual Languages*. Los Alamitos: IEEE Computer Society Press, 1995. 195–202. [doi: 10.1109/VL.1995.520809]
- [20] Janhunen T. On the intertranslatability of non-monotonic logics. *Annals of Mathematics and Artificial Intelligence*, 1999,27(1-4): 79–128. [doi: 10.1023/A:1018927416540]
- [21] Gottlob G. Translating default logic into standard autoepistemic logic. *Journal of the ACM*, 1995,42(4):711–740. [doi: 10.1145/210332.210334]

- [22] Imielinski T. Results on translating defaults to circumscription. *Artificial Intelligence*, 1987,32(1):131–146. [doi: 10.1016/0004-3702(87)90064-6]
- [23] Bottoni P, Taentzer G, Schürr A. Efficient parsing of visual languages based on critical pair analysis and contextual layered graph transformation. In: *Proc. of the 2000 IEEE Int'l Symp. on Visual Languages*. Los Alamitos: IEEE Computer Society Press, 2000. 59–60. <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=874351> [doi: 10.1109/VL.2000.874351]
- [24] Zou Y, Lü J, Zeng XQ, Ma XX, Yang QL. Constructing confluent context-sensitive graph grammars from non-confluent productions for parsing efficiency. In: Huang ML, Nguyen QV, Zhang K, eds. *Proc. of the Visual Information Communication*. New York: Springer-Verlag, 2009. 135–147. [doi: 10.1007/978-1-4419-0312-9_8]
- [25] Nagl M. Set theoretic approaches to graph grammars. In: *Proc. of the 3rd Int'l Workshop on Graph Grammars and Their Application to Computer Science*. LNCS 291, Berlin: Springer-Verlag, 1986. 41–54. <http://www.springerlink.com/content/ph04315m5r25w160/> [doi: 10.1007/3-540-18771-5_43]
- [26] Uesu T. A system of graph grammars which generates all recursively enumerable sets of labelled graphs. *Tsukuba Journal of Mathematics*, 1978,2(1):11–26.

附中文参考文献:

- [11] 曾晓勤,韩秀清,邹阳.一种基于边的上下文相关图文法形式化框架. *软件学报*,2008,19(8):1893–1901. <http://www.jos.org.cn/1000-9825/19/1893.htm> [doi: 10.3724/SP.J.1001.2008.01893]



邹阳(1976—),男,湖北武汉人,博士生,讲师,CCF 会员,主要研究领域为软件形式化方法,图文法,软件工程.



胡昊(1975—),男,博士,副教授,CCF 会员,主要研究领域为过程技术,软件过程,软件质量保障技术.



吕建(1960—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为软件形式化与自动化,面向对象方法与技术,软件工程与方法学.



宋巍(1981—),男,博士,讲师,CCF 会员,主要研究领域为服务计算,形式化方法,软件工程与方法学.



曹春(1978—),男,博士,讲师,CCF 会员,主要研究领域为网构软件,软件构件技术,软件自适应技术.



杨启亮(1975—),男,博士生,讲师,CCF 会员,主要研究领域为软件工程,自适应软件,控制系统软件.