

图语法综述^{*})

韩秀清¹ 曾晓勤¹ 邹阳¹ 张康²

(河海大学计算机与信息工程学院 南京 210098)¹

(德克萨斯大学达拉斯分校计算机科学系 德克萨斯 75080-3021)²

摘要 形式语言理论对计算机科学的发展起了重大的作用,作为对传统字符文法扩展的图文法的形式化研究,其重要意义是不言而喻的。本文在概述图文法的产生、发展和现状的基础上,着重介绍了从一维字符文法扩展到二维图文法所出现的新问题,以及在形式化处理上引出的新方法,其中最主要的是嵌入问题的解决、文法类型的划分和成员问题的判定。文中以目前较为流行的图文法为例,特别是一些典型的上下文无关和上下文相关的图文法,对上述的问题进行了深入的讨论,指出了现有方法中的一些不足之处,并展望了图文法今后值得研究的问题和方向。

关键词 形式语言,图文法,嵌入问题,文法类型,成员问题

Survey of Graph Grammars

HAN Xiu-qing¹ ZENG Xiao-qin¹ ZOU Yang¹ ZHANG Kang²

(Dept. of Computer Science & Engineering, Hohai University, Nanjing 210098, China)¹

(Dept. of Computer Science, The University of Texas at Dallas, Texas 75080-3021, USA)²

Abstract It is well known that formal language theory plays an important role in the development of computer science, and so will be the research on two dimensional graph grammar formalisms, which is an extension of one dimensional string grammars. Based on a summary of the generation, development and current status of graph grammars, this paper mainly introduces the newly emerging problems caused by the dimension extension and some new approaches, such as the solution to embedding problem, the classification of graph grammars and the decidability of membership problem. By taking some popular graph grammars as examples, especially some typical context-free and context-sensitive graph grammars, the paper discusses their approaches for solving the above mentioned problems, their shortcomings and unsolved research problems and future directions.

Keywords Formal language, Graph grammar, Embedding problem, Grammar classification, Membership problem

1 引言

自从乔姆斯基于1956年建立形式语言体系以来,形式语言理论取得了很大发展,对计算机科学产生了深刻影响,特别是在计算机语言等方面。但乔姆斯基形式化文法方法是针对一维字符文法而提出的,随着计算机软件技术的发展,特别是可视化人机界面的迅速发展,图、表等二维对象的使用更加广泛和深入,而乔姆斯基形式化文法方法对定义和分析这些二维对象已不能胜任,于是图文法作为一维文法的自然扩展应运而生。图文法最早的提出是为了解决图像处理问题,Pfaltz和Rosenfeld在1969年第一届IJCAI会议上发表了一篇题为“Web Grammars”的文章^[1],揭开了图文法研究的序幕。此后,人们在理论和应用方面进行了大量探索,在图文法形式化定义和图文法系统实现方面取得了许多成果^[2-6]。

在图文法研究中,任何二维对象,如图像和图形等,都被抽象为带有结点和边的图(有向图或无向图)来表示,其中结点用来表示对象元素,边用来表示图中结点之间的关系。与一维的字符文法的功能类似,图文法用来形式化定义和分析这类图;但与字符文法不同的是,其重点是解决在图的推导和

归约过程中可能产生悬边的嵌入问题,这是一维文法中所没有的。再就是解决较一维文法更加复杂的图文法表达能力(图文法的分类)和可判定性(归约过程的停机)等问题。

从检索文献可知,目前已有许多图文法及相关的形式化方法问世,其中多数是面向特定应用领域的,它们通常可化归为现有的两类图文法理论中的一种。这两类分别是基于代数的方法^[7]和基于集合的方法^[8]。二者分别以代数理论(范畴论)和集合论为基础来形式化定义和描述图文法的机制,证明有关的定理,以及设计归约算法等。近年来,图文法的研究采用基于集合的方法的居多,在对文法结构做一些限制之后研究了它们的性质,并给出了一些归约算法^[9,10]。另外,有人将基于代数的方法中的一些思想运用到图文法中,但不涉及用范畴论进行严格的数学定义,取得了一些比较好的效果^[11,12]。

在应用方面,图文法已被用来对多种图表进行识别和分析。Pfaltz提出了NRNTST系统,利用图文法对神经网络图像(neural network picture)进行分析^[13]。Janssens等人讨论了用图文法对示意图和流程图进行分析的方法^[14]。在此基础上,Fahmy等人又将图文法应用到音乐领域,用来对音符

^{*}) 本文工作得到国家自然科学基金(60571048、60673186)的资助。韩秀清 硕士研究生,主要研究方向是图文法及其应用;曾晓勤 教授,博导,主要研究方向是人工神经网络、图文法;邹阳 讲师,主要研究方向是图文法、非单调推理和逻辑程序;张康 教授,博导,主要研究方向是可视化语言、信息可视化和软件工程。

进行识别^[15];Dolado 用图文法来建造 Forrester diagram^[16], Göttler 用它来设计 CAD 系统^[17],进一步扩大了图语法在这一领域内的应用。此后,随着对图语法研究的逐渐深入,它被用来研究并发系统^[18]、进行数据流分析^[19]和软件体系结构描述^[20]、对 XML 文档进行分析和转换^[21]等。图语法的其它应用领域还包括图论、程序语言翻译、可视化语言、生物学、并发系统研究、数据库描述、增量编译程序、软件开发环境等,这些方面的应用详见文献^[22,23]。近年来,随着对上下文相关图语法研究的不断深入,人们开始研究利用它来对可视化语言进行形式化定义和语法分析^[24-26];另外,图语法在模式识别领域中也显示出良好的应用前景,目前有人用它来进行手写体的识别^[27]。

下面着重介绍从一维字符文法扩展到二维图语法所引出的新问题,以及在形式化处理上的一些方法,其中最主要的是嵌入问题的解决、文法类型的划分和成员问题的判定。以目前较为流行的图语法为例,特别是一些典型的上下文无关和上下文相关的图语法,对上述的问题进行深入的讨论。在指出了现有图语法存在的不足之处基础上,最后给出了作者在前期对图语法研究工作中总结出的一些观点和想法,供同行们研究参考。

本文下面章节的安排如下:第 2 节给出了图语法中的一些基本概念和符号的定义,并概述了图语法对图进行分析的方式和面临的一些问题,然后重点介绍了起决定作用的两方面:嵌入方法和产生式形式。第 3 节首先讨论了上述两方面在图语法中的配对的情况,然后对每一种情况分别给出了一个图语法的实例进行了介绍。最后进行了展望和总结。

2 图语法研究的主要问题及其分类

图语法是用来定义图语言和对图进行语法分析的形式化工具。图语法中的概念多源于图重写系统。所谓图重写,是指根据图重写规则对一个图进行变换而得到另一个图的过程。下面首先给出图重写中的一些常用定义。

2.1 图重写中的基本定义

定义 1 $G=(V, E, l_V, l_E, s, t)$ 称作是标识集 L_V, L_E 上的一个图,其中: V 是图的结点集,通常它由终结点集 V_T 和非终结点集 V_N 两部分组成; E 是图 G 的边集; $l_V: V \rightarrow L_V, l_E: E \rightarrow L_E$ 两个函数给出了图 G 中每个结点和边的标识; $s: E \rightarrow V, t: E \rightarrow V$ 两个函数给出了图 G 中每条边的端点(起点和终点)。

需要指出的是,定义 1 描述的是一个结点带有标识、边既有标识也有方向的图。对于不同的应用领域,图重写系统对于结点有无标识、边有无标识和属性以及有向或无向等约定会有所不同。

定义 2 图重写式是形如 $g_l \rightarrow g_r$ 的一个规则,其中 g_l 和 g_r 是两个图。规则指出如何对一个给定图进行重写,即找到和 g_l 同构的子图时,就可以用和 g_r 同构的图对其进行替换。

定义 3 主图(Host Graph)是使用图重写式来对其进行重写的图,记为 g^{host} 。另外, g^{host} 中和 g_l 同构的一个子图记为 g_l^{host} 。在图的重写过程中,它可以被和 g_r 同构的图所替换。在某些重写方式下, g_l^{host} 除了要和 g_l 同构,边也需要满足某些约束条件。

定义 4 g^{host} 中去除 g_l^{host} 的所有结点和边以及有一个端点落在 g_l^{host} 上的边之后所得到的图称作余图(Residual Graph),记为 $g^{residual}$ 。

定义 5 g_r^{host} 是和 g_r 同构、用来替换主图中的 g_l^{host} 的图。

图重写式是对主图进行重写的基本依据,但要完成主图的重写,仅有图重写式是不够的,还必须有相应的规则来对 g_r^{host} 如何嵌入到 $g^{residual}$ 中进行说明。这一规则称为嵌入规则,它和图重写式是进行图重写所必需的。在某些图重写系统中,图重写规则除了这两者之外,还包含了应用条件、属性转移函数等,但这些并不是图重写所必需的。

另外,在对图进行重写时,存在两种不同的重写方式:顺序重写和并行重写^[8]。前者每一步只在主图中寻找一个 g_l^{host} , 并进行替换;后者则在每一步重写时都将主图分成几部分分别进行重写。

2.2 从图重写到图语法

图重写可以完成图的变换,但它对图如何生成以及图结构的分析并不深究。与之相比,图语法则着重研究图的生成(推导)和图结构的分析(归约),使图的结构更加明显地呈现出来。图语法继承了图重写中的一些定义,但在另一些定义上图语法有更多的限制。下面是图语法中的几个新的定义。

定义 6 一个产生式 p 是在相同标识集上的一对图,通常用 (L, R) 或者 $L := R$ 表示,其中 L 和 R 分别称为产生式的左端和右端。

定义 7 一个图语法 gg 是一个三元组 (A, P, E) ,其中 A 是文法的初始图, P 是一组产生式, E 是文法的嵌入规则。

这个定义指出,图语法比图重写多了一个初始图。初始图是图语法对图进行变换的起点,文法中的所有图都是通过初始图进行变换得到的。另外,还与图重写不同的是,图重写规则的使用是立足于单向的图变换,而图语法产生式的使用则是兼顾双向的图生成和图识别。用图语法,既可以在主图中寻找和产生式左端同构且满足替换条件的子图,用和产生式右端同构的图对其进行替换;也可以寻找和产生式右端同构且满足替换条件的子图,用和产生式左端同构的图对其进行替换。这两个过程中,我们都把可替换的子图称为图柄(Redex)。前一过程称为推导,用它可以从初始图出发产生出一个图的集合,此集合称作是图语法所产生的语言。后一过程称为归约,用它可以判断一个图是否属于某个给定图语法产生的语言。在不严格区分推导和归约时,把它们统称为图的变换。

在对图进行变换的方式上,图语法一般都是采用的顺序方式,即每一步只寻找主图的一个图柄进行替换。在这种方式下,如果图 G 可以通过一步变换得到图 G' , 记作 $G \Rightarrow G'$ 。如果图 G 可以通过至少零步变换得到图 G' , 记作 $G \Rightarrow^* G'$ 。由此,我们可以定义图语法产生的语言。

定义 8 假设 $gg=(A, P, E)$ 是一个图语法, $L(gg)=\{G | A \Rightarrow^* G \wedge G$ 中不含非终结点\} 称作是该图语法产生的语言。

下面是一组图语法产生式和使用这组产生式进行推导的过程。有关它的嵌入规则将在下一节中进行详细介绍。

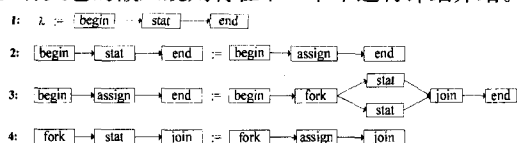


图 1 一组图语法产生式

图 1 是一个图语法的产生式集合,第一个产生式左端的 λ 是文法的初始图。我们使用这组产生式从初始图出发,可

以得到如图 2 所示的推导过程。

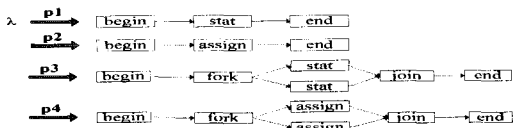


图 2 利用产生式对图进行变换过程

2.3 图文法的关键问题

如前所述,嵌入问题的解决、文法类型的划分和成员问题的判定是图文法中最关键的三个问题。其中嵌入问题是字符文法中所没有的,后两个问题则由于图的复杂结构而变得较字符文法更加复杂。

所谓嵌入问题,是指图柄被替换之时,替换图柄的部分如何嵌入到余图之中。这是文法从一维到二维所产生的新问题。在一维文法中,字符是以线性次序排列的,当运用产生式对字符串进行替换后,只要将替换字符串置于被替换字符串原来的位置,就可以得到一个新的线性排列的字符串,因而不存在嵌入问题。而文法从一维扩展到二维,语素间由线性排列关系变成了空间上的邻接关系,在进行图变换时,替换部分如何嵌入到余图之中必须要有详细的说明,以免形成悬边。所谓悬边,是指替换之前连接图柄与余图的边,这些边的一个端点在替换时随图柄从主图中删除,而又不知在嵌入的图中哪是新端点。为此,文法中必须给出确定新端点的方法,即嵌入规则。在一些图文法形式框架中,嵌入问题也被称作悬边问题。

在嵌入方法上,目前存在两种不同的描述方式:集合的方法和代数的方法。前者以集合论为理论基础,借助一些表达式来说明嵌入方法,这些表达式给出了替换部分需要和余图建立连接的结点,以及确定和这些结点相连的余图中结点所需要的信息^[8]。这些信息一般包括结点的标识,边的方向和标识,以及经过的路径等。代数方法则是用映射等一些数学概念来描述嵌入方法^[7]。

从具体的嵌入方法来看,目前已有的嵌入方法主要有下面列出的几类^[28],其中前几类都是采用集合的方法来描述的,最后一类最早出现在基于代数的方法的图文法中,采用代数方法进行描述,现在在一些图文法中借用了这种嵌入方法的思想但并不进行严格的数学描述。

(1)无限制型:这是采用集合的方法描述中最简单的一类,对表达式没有任何限制。

(2)方向和标识保留型:替换之后连接替换部分和余图的边都是根据替换前连接图柄和余图的边建立的,并且要求边的方向和标识保持不变。

(3)深度 1 型。要求替换之后余图中和替换部分的结点相邻的结点必须是在替换之前和图柄中的结点相邻的结点。

(4)简单型。必须同时满足上述(2)和(3)的条件。

(5)基本型。简单型,并且嵌入方法与余图中结点的标识无关。

(6)类似型。基本型,并且嵌入方法与替换前连接图柄和余图的边的方向和标识无关。

(7)不变型。产生式两端分别定义了一组图元(一般是结点),它们之间存在一一对应关系。当对图柄进行替换时,只要把每个图元的连接关系转移到其对应图元上即可。而除去这些图元之外,图柄的其它部分是不允许和余图直接相连的。

不同的图文法可能有不同的表达能力,图文法的表达能

力一方面与文法采用的嵌入方法相关,另一方面还与文法对产生式形式的约束相关。图文法对产生式形式的分类与字符文法类似,但更复杂一些,主要包括以下几类^[28]:

(1)无限制的产生式。对产生式的形式没有任何限制。

(2)单调的产生式。产生式左端的结点数小于或等于产生式右端的结点数。

(3)上下文相关的产生式。产生式左端的一部分是产生式右端的一个子图。

(4)上下文无关的产生式。产生式左端只有一个结点,且为非终结点。

(5)线型产生式。受限的上下文无关的产生式,产生式右端至多只含有一个非终结点。

(6)正则产生式。受限的上下文无关产生式,产生式右端要么全由终结点组成,要么最多有一个非终结点,并且其余的终结点都有一条边连接并指向这个非终结点。

与字符文法的分类不同,图文法的分类不仅取决于产生式形式而且还取决于嵌入方法。在两者之一确定的情况下,我们可以得到图文法之间表达能力的关系。比如,当产生式形式固定为上下文相关的时候,不同嵌入方法得到的图文法表达能力存在以下关系^[29]:不变型 \subseteq 类似型 \subseteq 基本型 \subseteq 简单型 \subseteq 深度 1 型,方向和标识保留型 \subseteq 无限制型。而在嵌入方法固定为无限制型的情况下,不同产生式形式的图文法表达能力之间存在如下关系^[29]:正则型 $=$ 线型 \subseteq 上下文无关型 $=$ 上下文相关型 $=$ 单调型 \subseteq 无限制型。

成员判定问题涉及两个层面的考虑。一方面,图文法的形式框架要保证归约的过程能在有限步完成,并给出肯定或否定的判定结果,也就是说,给定任意一个图,能够在有限时间内判断出此图是否属于给定文法产生的语言。这一点要通过定义适当的图文法形式予以保证。另一方面,仅保证停机还不够,还要求归约算法的复杂性不能太高,要能满足实际应用的要求。这方面是有许多值得深入研究的课题。

3 几种典型的图文法介绍

如前所述,一个图文法的表达能力由它的产生式类型和嵌入方法共同决定。上下文相关图文法和上下文无关图文法由于表达能力较强、产生式形式相对简单,在实际问题中的应用较为广泛。从它们和几类嵌入方法的关系来看,在采用前几类嵌入方法时,这两类图文法的表达能力是类似的。此时只需要研究形式更为简单的上下文无关图文法即可。而在采用不变型的嵌入方法时,上下文无关图文法的表达能力较之上下文相关图文法就弱了很多,因此这类嵌入方法往往应用于上下文相关图文法。下面将分别介绍上下文无关图文法和上下文相关图文法的典型形式框架,并对其归约算法进行分析。

3.1 上下文无关图文法

在图文法发展的初期,对采用前几类嵌入方法的上下文无关图文法的研究比较多,其中比较常见的有:NLC(Node Label Controlled)^[30]、NCE(Neighbourhood Controlled Embedding)^[31]、HRG(Hyperedge Replacement Grammar)^[32]、RG(Relational Grammar)^[33]、AMG(Attributed Multiset Grammar)^[34]、PLG(Picture Layout Grammar)^[34]等。

如前所述,上下文无关图文法的产生式左端是唯一的非终结点。这意味着在图的推导过程中,需要寻找的图柄也始终是一个唯一结点的形式。

NLC 文法和 NCE 文法是上下文无关图语法中比较典型的两种。它们所采用的嵌入方法都属于 2.3 节所提到的深度 1 型。在具体做法上,它们为文法建立了一组连接说明(connection instruction)来实现替换部分在余图中的嵌入。这里的连接说明是用相当于前面所提到的表达式给出的。这两种文法可以定义无向图或有向图,定义有向图的通常称为 ed-NLC 和 ed-NCE 文法。下面以 ed-NCE 文法为例对此类进行详细说明。

一个 ed-NCE 文法可以形式化描述为一个六元组 $(\Sigma, \Delta, \Gamma, \Omega, P, S)$ 。其中 Σ 是结点的标示集, $\Delta \subseteq \Sigma$ 是终结结点的标示集, Γ 是边的标示集, $\Omega \subseteq \Gamma$ 是终结边的标示集, P 是图文法的产生式集合, $S \in \Sigma - \Delta$ 是图文法的初始图。产生式 p 的形式是 $X \rightarrow (D, C)$, 其中 $X \in \Sigma - \Delta$, D 为产生式的右端, C 是连接说明。 C 的形式是 $(\mu, p/q, x, d)$, 其中 $\mu \in \Sigma$, $p, q \in \Gamma$, x 是 D 中的一个结点, $d \in \{in, out\}$ 。若 $d=out$, 那么在用和产生式右端 D 同构的图对主图中标示为 X 的非终结点(记作 v)进行替换时, 如果主图中原来从 v 出发经过一条标示为 p 的边可以到达某结点, 且该结点标示为 μ , 那么替换后, 要从替换图中的 x 结点引一条标示为 q 的边指向该结点。若 $d=in$, 则边的方向相反。这一描述是针对图的生成过程的, 在对图进行归约时可以等价地得到嵌入方法。

下面是一个 ed-NCE 文法的例子, 利用它进行推导可以得到所有的二叉树, 其中每个结点到其孩子结点有边相连, 每个叶子结点有一条边指向根结点。图 3 是文法的产生式, 其中 p_1 的连接说明为空, p_2 的连接说明为 $\{(\#, */*, z, in), (\#, r/r, x_1, out), (\#, r/r, x_2, out)\}$, p_3 的连接说明为 $\{(\#, */*, z, in), (\#, r/*, z, out)\}$, 连接说明中的 $\#$ 和 $*$ 代表没有标识的结点和边。图 4 是由它推导出一棵二叉树的过程。

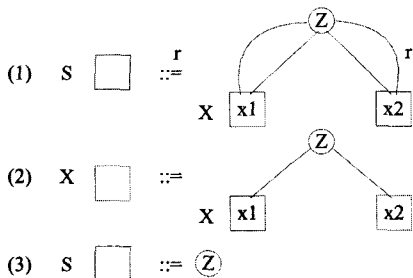


图 3 一组 ed-NCE 产生式

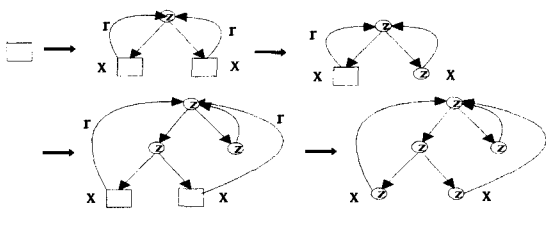


图 4 二叉树推导过程

与 NLC 文法和 NCE 文法相比, HRG 文法(超边替代图文法)在图的结构和产生式形式上都更复杂。HRG 文法中产生式的左端是一个超边(Hyperedge)。所谓超边, 实际上是一个带有固定数目触角(tentacle)的非终结点。触角是超边用来与周边结点建立连接的工具, 在超边所在的图中它的每一个触角都与某一结点相连。严格说来, 它已经不属于上下文无关图语法, 但超边的核心仍然是一个结点, 因此我们仍把它放在本节进行介绍。但由于它的结点附带了触角, 这种图文

法所采用的嵌入方法更类似于代数方法。

HRG 文法在每一个产生式的右端指定了某些结点为外部结点(external node), 并规定在使用产生式对主图进行变换之后, 只允许外部结点和余图中的结点建立相邻关系。同时, HRG 文法要求每个产生式左端超边所带的触角和右端的外部结点两者数目一致, 并且一一对应。HRG 文法就是借助这种一一对应关系来解决嵌入问题的。具体地说, HRG 文法在进行嵌入时是将主图中超边的触角所连接的结点和触角在产生式中对应的外部结点进行融合。所谓融合, 是指将两个结点合并为一个结点, 并将与每一个结点相关联的边与融合后的结点相关联。

图 5 是一个 HRG 产生式, 图 6 是利用它进行一步推导的过程。

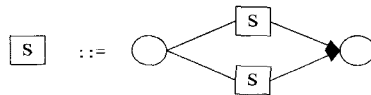


图 5 一个 HRG 产生式

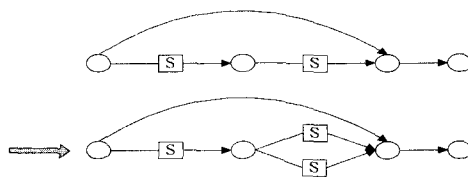


图 6 一步 HRG 推导过程

在解决成员判定问题时, 现有的归约算法针对的是满足合流(confluence)条件的 NLC 文法和 HRG 文法^[35]。所谓合流, 是指对于一个图文法来说, 当归约过程中同时出现多个图柄时, 无论先选取哪个进行归约最终得到的结果图都是相同的。若文法满足这一条件, 只要找到一个图柄, 就可以对其进行替换, 并且不需要回溯。这样一来, 算法的复杂度比较低, 但对合流条件的依赖使得这类方法的适用范围受到一些限制。

NLC 文法、NCE 文法、HRG 文法以及前面提到的 RG 文法、AMG 文法、PLG 文法等都属于上下文无关图语法, 在采用的嵌入方法足够复杂的情况下, 它们的表达能力能够等同于上下文相关图语法。但是, 在一些应用领域中, 存在难以用上下文无关图语法产生式直观描述的问题。比如在 ER 图(图 7)中, 经常会遇到下面形式的图的变换。



图 7 一个 ER 图转换实例

这是用上下文无关图语法无法直观地描述的。而用上下文相关图语法则可以直观地描述这一形式的变换。

3.2 上下文相关图语法

与上下文无关图语法相比, 上下文相关图语法的研究在图语法发展初期相对较少。1997 年, Rekers 针对在可视化语言中的应用提出了一种上下文相关分层图语法 LGG(Layered Graph Grammar)^[11]。此后, 人们对上下文相关图语法的研究逐渐多了起来, 又陆续出现了 RGG^[12]、SGG^[36] 等上下文相关图语法。

如前所述, 上下文相关图语法大多采用的是不变型的嵌入方法, 即在产生式的左右两端定义了一组一一对应的图元,

文法通过这些图元的对应关系来完成替换部分在余图的嵌入。

在 LGG 中,产生式两端的对应图元实际上是同一个结点。也就是说,在利用产生式进行替换时,它们并没有发生变化。LGG 文法将这些结点称作上下文,在产生式中进行了明确标注。

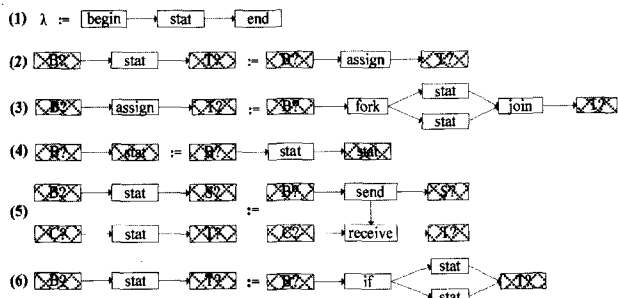


图 8 一组 LGG 产生式

图 8 是一组 LGG 的产生式。其中灰色的结点代表的是上下文。图中带有? 的结点或边的标识是一个通配符,所谓通配符,代表的是一组标识的集合。当图中某个结点或边的标识属于该集合时,就可以和该通配符代表的结点或边相匹配。比如我们可以将图 6 中的几个通配符定义为: $B? = C?$ = {begin, fork, if, stat}; $S? = T?$ = {end, assign, join, send, receive}。

在嵌入方法上, LGG 没有在产生式中给出嵌入规则,而是在图柄的定义中做了约束:除了满足和产生式左端或右端同构的条件之外,图柄中和产生式中的非上下文结点对应的结点不能和余图上的结点相邻。也就是说,在主图中图柄和余图的连接都是通过和产生式中的上下文结点相对应的结点来完成的。

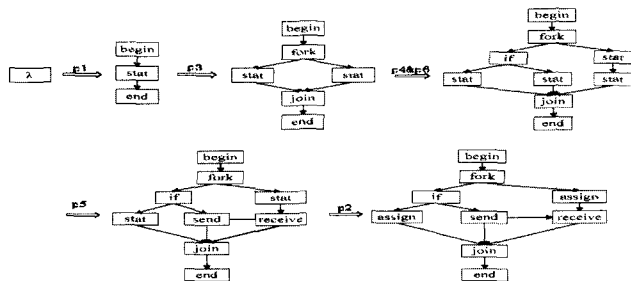


图 9 一个 LGG 推导过程

在这一条件下,在使用产生式对主图的图柄进行替换之后,只需将余图上和图柄有连接关系的结点连接到替换部分的对应上下文结点上就可以了,在此过程中不会出现悬边。图 9 描述了用图 8 中的文法推导得到某图的过程。

为了解决成员判定问题, LGG 文法通过对标识的分层来实现结点和边的分层,进而基于结点和边的层次关系定义了两图之间的“小于”关系,并要求产生式的左端小于右端,从而保证了文法的可判定性。

在归约算法上, LGG 和其它图元法有所不同。归约过程中,在将产生式左端的非上下文部分纳入到主图的同时,并不删除主图中的图柄,而是予以保留,然后继续对新图进行归约。规约算法^[37]包括自底向上和自顶向下两个阶段。在自底向上阶段,按照上述不删除图柄的方法对图进行归约,并记下每一步归约所用的产生式和归约前后图的形式,直到不再

有新的图柄出现为止。在这一步结束之后如果无法得到初始图,那么该图不属于相应的图元法。如果可以得到初始图,就继续进行自顶向下阶段的操作:首先找到从主图归约到初始图所用到的产生式和对应的归约前后图的形式,然后根据其依赖关系尝试创建这些产生式的应用顺序,若创建成功,它就对应图的归约过程,也就意味着该图属于相应图元法,反之说明它不属于该文法。

在 LGG 的这一归约算法中,第一个阶段中寻找图柄的过程是很复杂的, LGG 为此还专门设计了一个算法^[37]。而第二个阶段中创建产生式的应用顺序之前还要对它们之间的依赖关系进行判断。这进一步增加了 LGG 归约算法的复杂度。

综上所述, LGG 作为一种上下文相关图元法成功地解决了嵌入问题和成员判定问题,其思想对于上下文相关图元法的研究具有重要意义。但与此同时, LGG 的归约算法复杂度太高,使其很难应用到实际问题中。另外, LGG 对于通配符的使用也会带来一些问题^[12]。

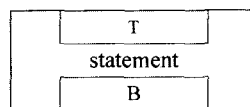


图 10 RGG 结点结构

为了解决 LGG 中面临的问题, Zhang 等人在 LGG 的基础上提出了 RGG 文法^[12]。RGG 继承了 LGG 的许多思想,但对图的结构进行了改变,并提出了一个复杂度较低的归约算法。RGG 文法对其所描述的图的结构进行了重新定义,其结点是一个包含一个超顶点(super vertex)和若干顶点(vertex)的二层结构,边的端点都落在具体的顶点或者超顶点上。图 10 所示为一个 RGG 文法的结点,该结点中 statement 标示了它的超顶点, T 和 B 分别标示了它的两个顶点。

为解决嵌入问题, RGG 中引入了一种称为标记(mark)的机制。所谓标记,实际上是赋给顶点的一个数值, RGG 图中的顶点可以有标记,也可以没有标记。当图中的顶点是有标记的时候,将其标记记作 $mark(v) = m$ 。RGG 要求同一个图上不同顶点的标记互不相同。实际上,这些标记顶点所起的作用与 LGG 中的上下文类似。正是在图结点中引入若干顶点以及在顶点上赋予标记的方法,使得 RGG 可以将上下文以带标记顶点的形式同时放入产生式两端,并只需引入一条公共的嵌入规则来给出替换方法。简单地说, RGG 要求图柄中与余图顶点相邻的都是带标记的顶点,并且每个产生式左端和右端中的标记顶点存在一一对应关系,这种关系由顶点的标记决定。当在主图中找到产生式一端(假设为左端)的图柄之后,首先将和产生式右端同构的一个图加到主图上;对于这一部分中的标记顶点,如果在图柄中存在一个顶点在产生式左端对应的顶点标记和它相同,那么就用图柄中的这个顶点替换它;最后从主图中删除图柄的其它部分。

如图 11 所示是一组 RGG 产生式,图 12 是利用它推导得到某图的过程。

在成员判定问题处理上, RGG 继承了 LGG 的分层方法,但在归约算法上, RGG 采用了 SFPA (selection-free parsing algorithm) 算法。与 LGG 的归约算法相比,这一算法的复杂度大大降低。但需要注意的是,并不是所有 RGG 文法下的图都可以使用 SFPA 算法,只有在满足一定条件时才可以使用该算法。Zhang 等人在文献^[12]中给出了一个可以使用该算法进行归约的条件,这一条件是对文法的产生式集合所作

的约束,称作是产生式集合的 selection-free 条件,其基本含义与 3.1 节提到的合流条件类似。

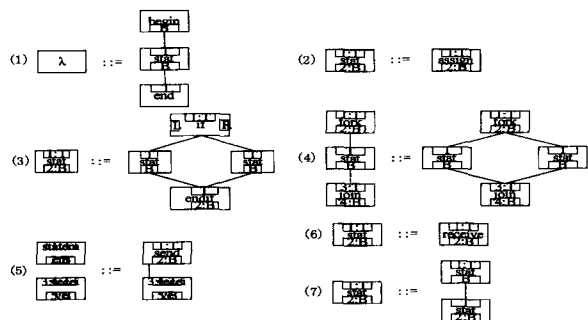


图 11 一组 RGG 产生式

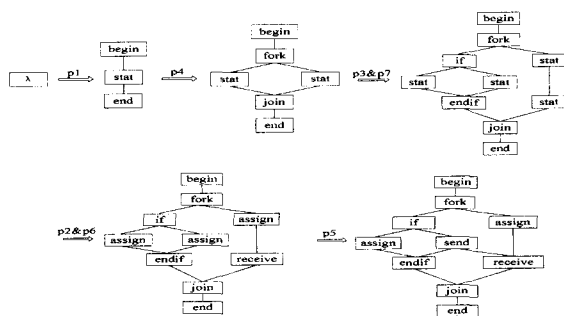


图 12 一个 RGG 推导过程

由于 RGG 给出了一个复杂度较低的归约算法,并避免了 LGG 使用通配符可能带来的问题,从而使得文法实用性更强。目前,该文法已经成功应用于 VisPro 可视化语言系统中^[24]。但 RGG 的 SFPA 算法的使用是有前提条件的。要满足这一条件,对于一般的文法使用者来说并不是一件容易的事。同时,RGG 保留了 LGG 对结点和边进行分层的思想,并且对图中的结点定义了双层结构,这使图的结构不够直观,在实际应用中如何定义双层结点成为新的问题。

为了简化 LGG 和 RGG 的分层方法,Zeng 等人对 RGG 进行改进,得到了 RGG+文法^[38]。RGG+ 摒弃了 LGG 和 RGG 对结点进行分层的思想,按照传统分类方式将结点分为终结点和非终结点,并基于此定义了图的大小,进而通过“产生式左边尺寸要小于或等于右边”的条件保证了文法的可判定性,这一条件还使得图的结构更加直观和便于使用。此外,RGG+ 给出了一个不依赖 selection-free 条件的归约算法,从而拓展了规约算法使用范围。但不可避免的是,该算法的复杂度与 SFPA 相比又大大提高。

之后,Kong 等人在 RGG 的基础上将空间信息加入到文法中来提高文法的表达能力,得到了 SGG^[36]。SGG 在定义中融合了对象间的抽象关系和空间关系,并在文法的分析中成功地使用空间关系信息降低了搜索图柄的复杂度。目前该文法也已在一些领域内得到应用^[39,40]。

从前面的描述可以看到,上下文无关图文法和上下文相关图文法作为目前应用最广泛的两类图文法,其研究已经取得了一些进展,但都还存在一些问题。

上下文无关图文法由于产生式的形式比较简单,为了满足表达能力上的需要,往往搭配比较复杂的嵌入方法使用。这些方法由于自由度很大,在制定具体的嵌入规则时对应背景背景的依赖性太强。另外,上下文无关的产生式在一些问题

的表达上也不够直观。上下文相关图文法则大多采用了不变型的嵌入方式。但从目前已有的上下文相关图文法来看,都还存在一些缺陷。比如 LGG 引入了通配符和层的概念,使得文法不直观且难设计,算法的复杂度也大大提高。RGG 由于图的结构不够直观,应用时需要从一般的图转换到 RGG 定义下的图。这些缺点使它们在实际应用时受到了很大限制。

结束语 众所周知,可视化语言具有表达直观、语义丰富、操作简便等诸多优点,在人机交互和软件开发等领域正扮演越来越重要的角色。如同传统字符语言的定义和实现需要字符文法的支持一样,图语言的定义和实现也需要有相应图文法的支持。遗憾的是,图语言虽然极大地方便了终端用户与计算机的交互,但却使得对它的形式化定义和分析变得十分困难和复杂,目前尚有许多需要解决的问题,主要有如下几个方面:①图文法的形式定义:现有的通用型图文法都太复杂,既不直观又不简便,更不利于自动生成和分析;②悬边的处理:它是文法由一维扩展到二维引起的,也是导致图文法不直观简便的主要根源,现有的嵌入方法与语义有太多的牵连,使得对图文法的设计如同编程一样,这与引入形式文法方法的初衷相悖;③复杂性的研究:本来在处理较自由的一维文法的语言时就会面临指数级的计算复杂性,维数升高后,复杂性跟着升高是难免的,这大大降低图文法的实用性。

上述的三个问题实际上是相互关联的,值得深入研究。针对它们,这里给出我们在前期摸索中总结出的一些想法:①充分运用形式化机制(如推导和归约等)来简化图文法产生式规则的表达,例如对复杂的图,考虑用简单的产生式规则借助多次推导或归约来产生或分析,而避免像图重写那样用复杂的规则;②抽象和归纳出嵌入过程中的结构信息,考虑将嵌入方法也通过产生式或类似于产生式的机制来形式化表示,尽可能为进一步研究图文法的自动化生成铺路;③从多方面考虑复杂性的因素,如设置适当限制条件对图文法分类来掌控复杂性,对诸如子图匹配、产生式选择、减少回溯、子图替换等方面寻找降低复杂性的算法。

总之,图文法的研究是计算机科学技术在二维图形图像空间发展的需要,更是实现可视化人机友好界面的理论基础,有着光辉的前景,但也面临着高复杂性的挑战。

参 考 文 献

- [1] Pfaltz J, Rosenfeld A. Web Grammars//Proc. First International Joint Conference on Artificial Intelligence. Washington, 1969
- [2] Rozenberg G. Handbook of Graph Grammars and Computing by Graph Transformation. Volume 1. Singapore: World scientific Publishing, 1997
- [3] Ehrig H, Engels G, Kreowski H-J, et al. Handbook of Graph Grammars and Computing by Graph Transformation: Applications, languages and Tools. Singapore: World scientific Publishing, 1999, 2
- [4] Ehrig H, Kreowski H-J, Montanari U, et al. Handbook of Graph Grammars and Computing by Graph Transformations; Concurrency, parallelism, and Distribution. Singapore: World scientific Publishing, 1999, 3
- [5] Drewes F, Hoffmann B, Janssens D, et al. Adaptive Star Grammar. ICGT, 2006: 77-91
- [6] Lara J, Bardohl R, Ehrig H, et al. Attributed Graph Transformation with Node Type Inheritance. Theoretical Computer Science, 2007, 376(3): 139-163

- [7] Ehrig H. Introduction to the Algebraic Theory of Graph Grammars// Claus V, Ehrig H, Rozenberg G, eds. Graph-Grammars and their Application to Computer Science and Biology, Berlin: Springer-Verlag, 1979:1-69
- [8] Nagl M. A tutorial and bibliographical survey on graph-grammars// Claus V, Ehrig H, Rozenberg G, eds. Graph-Grammars and their Application to Computer Science and Biology, Berlin: Springer-Verlag, 1979:70-126
- [9] 花全香,邢汉承,冯纯伯. 一种有效的结点符号上下文无关图文法分析算法. 软件学报, 1995, 6(9):567-575
- [10] 孟祥武. 图文法遗传算法. 计算机工程与科学, 1998, 20(4):11-16
- [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 D, Zhang K, Cao J. A Context-sensitive Graph Grammar Formalism for the Specification of Visual Languages. The Computer Journal, 2001, 44(3):187-200
- [13] Pfaltz J. Web Grammars and Picture Description. Computer Graphics and Image Processing, 1972, 1(1):193-220
- [14] Bunke H. Attributed Programmed Graph Grammars and Their Application to Schematic Diagram Interpretation. IEEE Pattern Analysis and Machine Intelligence, 1982, 4(6):574-582
- [15] Fahmy H, Blostein D. A Graph-Grammar Programming Style for Recognition of Music Notation. Machine Vision and Applications, to appear 1992 Preliminary results// Proc. First International Conference on Document Analysis & Recognition. St. Malo, France, Sept. 1991:70-78
- [16] Dolado J, Torrealdea F. Formal Manipulation of Forrester Diagrams by Graph Grammars. IEEE System, Man and Cybernetics, 1988, 18(6):981-996
- [17] Göttler H, Günther J, Nieskens G. Use Graph Grammars to Design CAD-System// Fourth International Workshop on Graph Grammars and Their Application to Computer Science:396-410
- [18] 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
- [19] 花全香,邢汉承,冯纯伯. 数据流分析的图文法途径. 小型微型计算机系统, 1997, 18(7):42-48
- [20] 石兵,冉平,马晓星,等. 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
- [21] Taentzer G, Carughi G T. A Graph-based Approach to transform XML Documents. Fase, 2006:48-62
- [22] Third International Workshop on Graph Grammars and Their Application to Computer Science// Ehrig H, Nagl M, Rozenberg G, et al., eds. Lecture Notes in Computer Science, Vol. 291 Springer Verlag, 1987
- [23] Fourth International Workshop on Graph Grammars and Their Application to Computer Science// Ehrig H, Kreowski H, Rozenberg G, eds. Lecture Notes in Computer Science, Vol. 532 Springer Verlag, 1991
- [24] Zhang K, Zhang D, Cao J. Design, Construction, and Application of A Generic Visual language Generation Environment. IEEE Trans. Software Eng., 2001, 27:289-307
- [25] Amaral V, Helmer S, Moerkotte G. Formally Specifying the Syntax and Semantics of a Visual Query Language for the Domain of High Energy Physics Data Analysis// Proc. 2005 IEEE Symposium on Visual Languages and Human-Centric Computing. Dallas, USA, September 2005:251-258
- [26] Ates K L, Kukluk J, Holder L, et al. Graph Grammar Induction on Structural Data for Visual Programming// Proc. 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06). Washington D. C., USA, IEEE CS Press, November 2006:232-239
- [27] Blostein D, Schürr A. Visual Modeling and Programming with Graph Transformations// Tutorial at 14th IEEE Symp. On Visual Languages. Halifax, NS, Canada, Sep, 1998
- [28] Fahmy H, Blostein D. A Survey of Graph Grammars: Theory and Applications. Pattern Recognition, 1992, 2:294-299
- [29] Blostein D, Fahmy H. Practical Use of Graph Rewriting. Technical Report// The 5th International Workshop on Graph Grammars and Their Application to Computer Science. 1994
- [30] Rozenberg G, Welzl E. Boundary NLC Graph Grammar—Basic Definition, Normal Forms, and Complexity. Information and Control, 69:136-167
- [31] Janssens D, Rozenberg G. Graph Grammars with Neighbourhood-controlled Embedding. Theoretical Computer Science. 1982 (21):55-74
- [32] Drewes F, Kreowski J, Habel A. Hyperedge Replacement Graph grammars. In[2]:95-156
- [33] Wittenburg K. Earley-Style Parsing for Relational Grammars// Proc. 8th IEEE Workgroup on Visual Languages. September, Seattle, WA, Los Alamitos, CA: IEEE Computer Society Press, 192-199
- [34] Golin E. A Method for the Specification and Parsing of Visual Languages. PhD Thesis. Dept. of computer science, Brown University, 1991
- [35] Engelfriet J, Rozenberg G. Node Replacement Graph Grammars. In[2]:1-94
- [36] Kong J, Zhang K, Zeng X. Spatial Graph Grammars for Graphical User Interfaces. ACM Transactions on Computer-Human Interaction, 2006, 13(2):268-307
- [37] Rekers J, Schürr A. A Parsing Algorithm for Context-sensitive Graph Grammar. Technical Report 95-05. Leiden University, the Netherlands
- [38] Zeng X, Zhang K, Kong J, et al. RGG+: An Enhancement to the Reserved Graph Grammar Formalism// Proceedings of the 2005 IEEE Symposium on Visual Languages and Human-Centric Computing. 2005:272-274
- [39] Zhang K, Kong J, Qiu M, et al. Multimedia Layout Adaptation Through Grammatical Specifications. ACM/Springer Multimedia Systems, 2005, 10(3):245-260
- [40] Zhao C, Kong J, Dong J, et al. Pattern-based Design Evolution Using Graph Transformation. Journal of Visual Languages and Computing, 2007, 18(4):378-398