一种有效的结点标号上下文无关图文法分析算法*

花全香 邢汉承 冯纯伯

(东南大学计算机系,南京 210018)

摘要 本文讨论了上下文无关图文法的性质,并证明了图文法推导具有独立性.本文还给出了一种有效的上下文无关图文法分析算法,它具有多项式时间复杂性,并给出了算法的正确性证明.该算法已经用C语言实现.

关键词 上下文无关图文法,文法分析.

图文法是形式语言理论从一维文法(串文法)到二维文法的自然推广,在句法模式识别、图象处理、程序设计语义学等方面具有广泛的应用前景.在过去几十年中,对基于二维甚至多维结构的形式文法在理论上和应用上进行了大量的研究,在图文法和图文法重写系统中取得了不少进展[1,2].在目前已经给出的图文法系统中,由于所面向的应用领域的不同,给出了多种不同的上下文无关图文法的定义,而图文法的定义不同,它们的性质也将不同,甚至相矛盾.在本文中,我们所研究的图文法是基于结点标号的上下文无关图文法.文法分析是形式语言理论中的一个最为主要的研究方向.由于图不存在一种固定的扫描规则,来确定相应的产生式规则,因此图文法分析是比较困难的.目前已经提出的几种图文法分析算法或者是指数时间复杂性[3,4],或者为了能有一个有效的时间复杂性,而对图文法加了许多约束条件,如算符优先图文法[5,6].本文提出了一种有效的图文法分析算法,它具有多项式时间复杂性.该算法是一个自上而下的图文法分析算法.另外本文还证明了上下文无关图文法的推导过程是相互独立的.

1 术 语

图文法是串文法的自然推广,是串文法的一个一般化描述,如果用结点来表示基本模式元或中间模式元,用边表示它们之间的关系,则一个模式可抽象地描述为一个图.

定义 1.1. 一个定义在 Σ(结点标号符号表)上的图 G,是一个三元组 $G=(N,E,\varphi)$,其

* 本文1993-09-14收到,1994-04-26定稿

作者**花全香**,1964年生,讲师,主要研究领域为人工智能,模式识别.**邢汉承**,1938年生,教授,主要研究领域为人工智能,逻辑程序设计,计算机应用.**冯纯伯**,1928年生,教授,主要研究领域为系统建模,自适应鲁棒及智能化控制理论及应用.

本文通讯联系人:花全香,南京 210018,东南大学计算机系

中 N 是一个有穷非空的结点集，$E\subseteq N\times N$ 是图 G 的边集，$\varphi: N\to\Sigma$ 是结点标号函数. 为讨论方便，用 N_G, E_G, φ_G 分别表示图 G 的结点集、边集和结点标号函数.

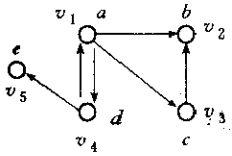

图1

例 1：$G=(N_G,E_G,\varphi_G)$，$N_G=\{v_1,v_2,v_3,v_4,v_5\}$，

$\varphi_G(v_1)=a,\varphi_G(v_2)=b,\varphi_G(v_3)=c,\varphi_G(v_4)=d$

$\varphi_G(v_5)=e, E_G=\{\langle v_1,v_2\rangle,\langle v_1,v_3\rangle,\langle v_3,v_2\rangle,$

$\langle v_4,v_1\rangle,\langle v_4,v_5\rangle\}$，它的图形表示如图 1.

图的形式表示和图的拓扑表示是图的二种表达方式，本文将根据需要，不加区分地使用.

定义 1.2. $G=(N_G,E_G,\varphi_G)$ 与 $H=(N_H,E_H,\varphi_H)$ 是定义在 Σ 上的两个图，称 G 等价于 $H(G\equiv H)$ 当且仅当存在一个双射函数 f，$f: N_G\to N_H$，且满足：

(1) $\varphi_G(m)=\varphi_H(f(m))$，对 $\forall\ m\in N_G$.

(2) $\langle m_1,m_2\rangle\in E_G$ 当且仅当 $\langle f(m_1),f(m_2)\rangle\in E_H$.

对任意两个图 G_1 和 G_2，通过等价变换，总能得到 $N_{G_1}\cap N_{G_2}=\Phi$. 在本文中，除非有特别说明，总是假定图的结点集两两不相交.

定义 1.3. 设 N 是一个集合，记 $\#N$ 为集合 N 的势.

定义 1.4. 一个上下文无关图文法 $CFGG$ 是一个四元组，$GG=(\Sigma_n,\Sigma_t,S_o,R)$，$\Sigma_n$ 是图文法的非终极结点符号表，Σ_t 是图文法的终极结点符号表，$\Sigma_n\cup\Sigma_t=\Sigma$，$S_o\in\Sigma_n$ 为图文法的初始符号，R 是一个有穷非空的产生式规则集，R 中的每个规则是一个四元组，$P=(A,D,B_{in},B_{out})\in R$，它满足：

(1) $A\in\Sigma_n$ 是一个非终极结点符号，称之为产生式规则的左部(*left—hand—side*)，亦记 $lhs(P)=A$.

(2) $D=(N_D,E_D,\varphi_D)$ 是一个定义在 Σ 上的图，称之为产生式规则的右部(*right—hand—side*)，亦记作 $rhs(P)=D$.

(3) $B_{in}\subseteq N_D, B_{out}\subseteq N_D$ 分别为图 D 中的结点子集，称之为产生式规则的入口结点集和出口结点集，亦分别记作 $B_{in}(P), B_{out}(P)$.

定义 1.5. $G_1=(N_{G_1},E_{G_1},\varphi_{G_1})$ $G_2=(N_{G_2},E_{G_2},\varphi_{G_2})$ 是定义在 Σ 上的两个图，$P=(A,D,B_{in},B_{out})\in R$ 是图文法的一个产生式规则，$m\in N_{G_1}$ 是图 G_1 中的一个结点，且 $\varphi_{G_1}(m)=A=lhs(P)$. 称图 G_2 是图 G_1 对结点 m 应用产生式规则 P 的一个直接推导. 在不引起混淆的情况下，简记作：$G_1(m)\overset{P}{\Rightarrow}G_2$ 或 $G_2=G_1(m(A)\leftarrow D)$，当且仅当：

(1) $N_{G_2}=(N_{G1}-\{m\})\cup N_D$.

(2) 对 $\forall\ n\in N_{G_2}$，$\varphi_{G_2}(n)=\begin{cases}\varphi_{G_1(n)} & \text{如果 } n\in N_{G_1}-\{m\}.\\ \varphi_{D(n)} & \text{如果 } n\in N_D.\end{cases}$

(3) $E_{G_2}=\{\langle n_1,n_2\rangle\}\,|\,n_1,n_2\in N_{G_1}-\{m\},\langle n_1,n_2\rangle\in E_{G_1}\}$

$\cup\{\langle n_1,n_2\rangle\}\,|\,n_1,n_2\in N_D,\langle n_1,n_2\rangle\in E_D\}$

$\cup\{\langle n_1,n_2\rangle\}\,|\,n_1\in N_{G_1}-\{m\},n_2\in N_D,n_2\in B_{in}(P),\langle n_1,m\rangle\in E_{G_1}\}$

$\cup\{\langle n_1,n_2\rangle\}\,|\,n_1\in N_D,n_2\in N_{G_1}-\{m\},n_2\in B_{out}(P),\langle m,n_1\rangle\in E_{G_1}\}$

例 2：图 G_1 中的结点 v_3(图 2)应用产生式规则 P(图 3)，可推导出图 G_2(图 4).

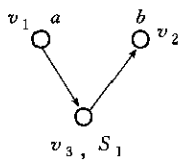

图2

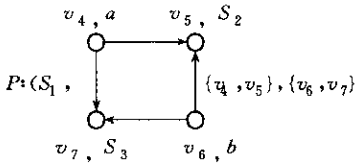

图3

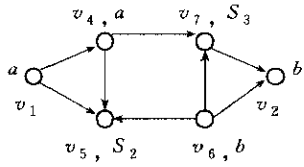

图4

定义 1.6. 由上下文无关图文法的初始结点 S_0,借助于一系列产生式的应用可推导出来的图 H 称之为图型,即 $S_0 \overset{*}{\Rightarrow} H$. 若图型 H 中所有结点标号均为终极结点符,则称图型 H 为上下文无关图文法的一个句子. 图文法 GG 的所有句子的集合称之为图文法的语言. $L(GG)=\{H \mid S_0 \overset{*}{\Rightarrow} H, \forall\ n \in N_H, \varphi_H(n) \in \Sigma_t\}$.

称图文法 GG 和 GG' 是等价的,当且仅当 $L(GG)=L(GG')$.

定义 1.7. 设图 G 是上下文无关图文法的一个图型,即存在一推导序列 τ, $S_0 \overset{\tau}{\Rightarrow} G$,则推导序列 τ 可以用树来表示,称之为图型 G 的文法推导树 T.

从 τ 构造推导树 T 的过程如下:

(1)若在 τ 中存在一产生式 P_i, $S_0=lhs(P_i)$,则 P_i 为树 T 的根结点.

(2)对树 T 中的任一中间结点 $P_i \in \tau$,,若 P_i 的右部包含非终极结点符 $S_{j_1}, \cdots, S_{j_k}$,相应地在 τ 中有 $P_{j_1}, \cdots, P_{j_k}$,分别有 $lhs(P_{j_1})=S_{j_1}, \cdots, lhs(P_{j_k})=S_{j_k}$,则结点 P_i 有 k 个子结点,这 k 个子结点分别为 $P_{j_1}, \cdots, P_{j_k}$. 若 P_i 的右部不含任何非终极结点符,则该结点 P_i 称为树的叶子结点.

由定义知,图文法的推导树与串文法的推导树不同. 在串文法的推导树中,结点的标号是文法的终极符或非终极符. 而在图文法推导树中,结点的标号是图文法的产生式规则. 因此图文法的推导树并不反映每一步推导的图型的演变过程,而只反映了图型 G 的一个推导过程.

2　图文法的数学特性

本节首先研究上下文无关图文法的一些基本数学特性,从中并可以看出:上下文无关图文法对每个非终极结点标号的推导是相互独立的. 后面将给出相应的图文法分析算法.

引理 2.1. 上下文无关图文法的重写规则满足交换律. 即对一个图型 G_0, G_0 中存在结点 $n_1, n_2 (n_1 \neq n_2)$,且 $\varphi_{G_0}(n_1)=S_1 \in \Sigma_n$, $\varphi_{G_0}(n_2)=S_2 \in \Sigma_n$,在图文法的产生式规则集 R 中,存在产生式规则 $P_1=(S_1, D_1, B_{in}^1, B_{out}^1)$, $P_2=(S_2, D_2, B_{in}^2, B_{out}^2)$,有 $G_0(n_1) \overset{P_1}{\Rightarrow} G_1(n_2) \overset{P_2}{\Rightarrow} G_2$, $G_0(n_2) \overset{P_2}{\Rightarrow} G_3(n_1) \overset{P_1}{\Rightarrow} G_4$,则 $G_2 \equiv G_4$,即 $G_0(n_1(S_1) \leftarrow D_1)(n_2(S_2) \leftarrow D_2) \equiv G_0(n_2(S_2) \leftarrow D_2) n_1(S_1 \leftarrow D_1)$.

证明:略,详见文献[7].

定义 2.1. 文法推导规则的结合$\otimes$,设 $P_1, P_2 \in R$ 是上下文无关图文法 GG 的两个产生式规则, $P_1=(S_1, D_1, B_{in}^1, B_{out}^1)$, $P_2=(S_2, D_2, B_{in}^2, B_{out}^2)$. 存在 $n_1 \in N_{D_1}$, $\varphi_{D_1}(n_1)=S_2$,记 $D_1(n_1) \overset{P_2}{\Rightarrow} D_3$,则产生式规则 $P_3=(S_1, D_3, B_{in}^3, B_{out}^3)$ 是产生式 P_1, P_2 的结合,记 $P_3=P_1 \otimes P_2$,其中

$$B_{in}^3=\begin{cases}B_{in}^1 & \text{如果 } n_1\notin B_{in}^3.\\ (B_{in}^1-\{n_1\})\cup B_{in}^2 & \text{如果 } n_1\in B_{in}^3.\end{cases}\qquad B_{out}^3=\begin{cases}B_{out}^1 & \text{如果 } n_1\notin B_{out}^3.\\ (B_{out}^1-\{n_1\})\cup B_{out}^2 & \text{如果 } n_1\in B_{out}^3.\end{cases}$$

引理 2.2. 上下文无关图文法的重写过程满足结合律. 即对一个图型 G_0,G_0 中存在一个结点 $n_1\in N_{G_0}$,且 $\varphi_{G_0}(n_1)=S_1$,存在产生式 $P_1=(S_1,D_1,B_{in}^1,B_{out}^1)\in R$,在图 D_1 中存在结点 $n_2\in N_{D_1}$,$\varphi_{D_1}(n_2)=S_2$,存在产生式 $P_2=(S_2,D_2,B_{in}^2,B_{out}^2)\in R$,记 $P_3=P_1\otimes P_2$,存在推导 $G_0(n_1)\overset{P_1}{\Rightarrow}G_1(n_2)\overset{P_2}{\Rightarrow}G_2$,$G_0(n_1)\overset{P_3}{\Rightarrow}G_3$,则 $G_2\equiv G_3$,即 $G_0(n_1(S_1)\leftarrow D_1)(n_2(S_2)\leftarrow D_2)\equiv G_0(n_1(S_1)\leftarrow D_1(n_2(S_2)\leftarrow D_2))$.

证明:略,详见文献[7].

定理 2.3. 设 GG 是一个上下文无关图文法,$S\in\Sigma_n$ 是一非终极结点符号,G 是一个有向图,$S_1\overset{+}{\Rightarrow}G_1$,当且仅当存在一个图 H,$S_1,\cdots S_m$ 是图 H 的非终极结点符号,$G_1,\cdots,G_m$ 是有向图,且满足:

(1)$S\overset{1}{\Rightarrow}H$,(即 H 是从 S 应用一个产生式规则所导出的).

(2)$S_i\overset{+}{\Rightarrow}G_i$,$i=1,2,\cdots,m$.

(3)$G=H(S_1\leftarrow G_1,S_2\leftarrow G_2,\cdots,S_m\leftarrow G_m)$.

证明:($\Diamond$)设 $S\overset{i}{\Rightarrow}G$,对推导步数 i 作归纳证明:

若 $i\geqslant 2$,$S\overset{i}{\Rightarrow}G$,可分解成两步推导,$S\overset{j}{\Rightarrow}K$,$A\overset{l}{\Rightarrow}L$,$j+l+1=i$,且在图 K 中,存在结点 $n\in N_k$,$\varphi_k(n)=A$,且 $G=K(n(A)\leftarrow L)$.

由归纳假设,对 $S\overset{j}{\Rightarrow}K$,存在图 H 与 m 个非终极结点符 $S_1,\cdots S_m$,及图 $G,\cdots,G_m$ 满足条件(1)(2)(3).

分两种情况:

①存在结点 $n\in N_H$,$\varphi_H(n)=A$,即在图 H 中存在非终极结点符 A,则 $G=K(n(A)\leftarrow L)=H(S_1\leftarrow G_1,\cdots,S_m\leftarrow G_m,A\leftarrow L)$.

②在图 H 中,没有满足条件①的结点,则必存在一图 G_j,存在 $n\in N_{G_j}$,$\varphi_{G_j}(n_1)=A$,不失一般性,不妨假设图 G_1 满足该条件,则 $G=K((n)A\leftarrow L)=H(S_1\leftarrow G_1(m(A)\leftarrow L),\cdots,S_m\leftarrow G_m)$. 显然必要性成立.

($\Diamond$)对图 H 中非终极结点符个数 m 进行归纳证明.

记 $K=H(S_2\leftarrow G_2,\cdots,S_m\leftarrow G_m)$,则由归纳假设 $S\overset{+}{\Rightarrow}K$,又 $G=H(S_1\leftarrow G_1,\cdots,S_m\leftarrow G_m)=K(S_1\leftarrow G_1)$.

因此,$K\overset{1}{\Rightarrow}G$,即有 $S\overset{+}{\Rightarrow}G$. 充分性成立.

直观地,上下文无关图文法的推导对不同的非终极结点符导出的子图相互独立. 因此对它们的推导过程可独立地对待,同样对上下文无关图文法的分析过程来说,归约可独立对待.

设 S 是上下文无关图文法的非终极结点符,G 是一个有向图.

算法 2.1. 图 G 的推导算法 $Derive(S,G)$.

(1)若 $S\overset{t}{\Rightarrow}G$,则返回 $True$.

(2)若对一个图 H,$N_H \subseteq N_G$,存在 $n_1,\cdots,n_m \in N_H$,$\varphi_H(n_i)=S_i(i=1,\cdots,m)$,$G_1,\cdots,G_m$ 是图,且满足 $S \overset{1}{\Rightarrow} H$,$G=H(S_1 \leftarrow G_1,\cdots,S_m \leftarrow G_m)$,并且对所有 $i=1,\cdots,m$ 有 $Derive(S_i,G_i)=True$,则返回 $True$,否则返回 $False$.

(3)返回 $False$.

算法 2.2. 图 G 的分析算法 $Parsing(S,G)$

(1)若 $S \overset{1}{\Rightarrow} G$,则返回 $S \overset{1}{\Rightarrow} G$.

(2)若对一个图 H,$N_H \subseteq N_G$,存在 $n_1,\cdots,n_m \in N_H$,$\varphi_H(n_i)=S_i(i=1,\cdots,m)$,$G_1,\cdots,G_m$ 是图,且满足 $S \overset{1}{\Rightarrow} H$,$G=H(S_1 \leftarrow G_1,\cdots,S_m \leftarrow G_m)$,并且对所有的 $i=1,\cdots,m$,有 $Parsing(S_i,G_i) \neq \perp$,则返回:$(S \overset{1}{\Rightarrow} H,(S_1,Parsing(S_1,G_1)),\cdots,(S_m,Parsing(S_m,G_m)))$,否则返回 $\perp$.

(3)返回 $\perp$.

其中 $\perp$ 表示图 G 不可能由 S 推导出来,即分析失败.

由于算法中存在递归调用,因此算法 2.1,2.2 的时间复杂性为指数阶.

3　一种有效的图文法分析算法

本文所提出的上下文无关图文法分析算法是一种自上而下的分析算法(与串文法的 Earley 算法类似[8]),该算法从被分析的图 G 中依次读入结点,并为每个结点构造一个分析表,从而实现对图的文法分析.

为了方便后面的讨论,对图 $G=(N_G,E_G,\varphi_G)$,如果存在一结点 $n \in N_G$,$\varphi_G(n)=a$,则简记作 $a \in G$. 如果存在结点 $n_1,n_2 \in N_G$,$\varphi_G(n_1)=a$,$\varphi_G(n_2)=b$,且 $(n_1,n_2) \in E_G$. 则记 $a \rightarrow b \in G$. 设 $P=(S,D,B_{in},B_{out})$ 是图文法的产生式,如果存在结点 $n \in N_D$,$\varphi_D(n)=a$,若 $n \in B_{in}$,则记 $a \in B_{in}(P)$,若 $n \in B_{out}$,则记 $a \in B_{out}(P)$.

设 I' 是这样一个产生式规则集合,它只有唯一的一个产生式规则 P_i 没有被标记为 $Used$. 对 I' 中的任一产生式 P_j,在 P_j 的右部所有被标记为 $Used$ 的非终极结点符 $X \in rhs(P_j)$,在 I' 中存在相应的被标记为 $Used$ 的产生式 P_k,$X=lhs(P_k)$.

定义 3.1. $First(I')$ 是这样一个集合:

(1)记 $P=P_i$,P_i 是 I' 中没有被标记的产生式.

(2)对所有 $x \in B_{in}(P)$,x 被标记为 $Used$.

①如果 $x \in \Sigma_t$,则 $x \in First(I')$.

②如果 $x \in \Sigma_n$,存在相应的产生式 $P_j \in I'$,$x=lhs(P_j)$,令 $P=P_j$,重复过程(2).

定义 3.2. $Last(I')$ 是这样一个集合:

(1)记 $P_i=P$,P_i 是 I' 中没有被标记为 $Used$.

(2)对所有 $x \in B_{out}(P)$,x 记为 $Used$.

①如果 $x \in \Sigma_t$,则 $x \in Last(I')$.

②如果 $x \in \Sigma$,存在相应的产生式 $P_j \in I'$,$x=lhs(P_j)$,令 $P=P_j$,重复过程(2).

设 P_k 是 I' 的一个产生式,$S_k \in rhs(P_k)$,且 S_k 在 P_k 中没有被标记为 $Used$.

定义 3.3. $Before(S_k,P_k,I')$ 是这样一个集合:

对所有 $x \rightarrow S_k \in rhs(P_k)$，$x$ 被标记为 $Used$.

(1)如果 $x \in \Sigma_t$，则 $x \in Before(S_k, P_k, I')$.

(2)如果 $x \in \Sigma_n$，则在 I' 中存在相应的产生式 P_3, P_j 为已标记的产生式，且 $x = lhs(P_j)$，对所有 $y \in B_{out}(P_j)$，y 被标记为 $Used$.

①如果 $y \in \Sigma_t$，则 $y \in Before(S_k, P_k, I')$.

②如果 $y \in \Sigma_n$，令 $x = y$，转(2).

定义 3.4. $Fellow(S_k, P_k, I')$是这样一个集合：

对所有 $S_k \rightarrow x \in rhs(P_k)$，$x$ 被标记为 $Used$.

(1)如果 $x \in \Sigma_t$，则 $x \in Fellow(S_k, P_k, I')$.

(2)如果 $x \in \Sigma_n$，则在 I' 中存在相应的产生式 P_i, P_j 为已标记的产生式，且 $x = lhs(P_j)$，对所有 $y \in B_{in}(P_j)$，y 被标记为 $Used$.

①如果 $y \in \Sigma_t$，则 $y \in Fellow(S_k, P_k, I')$.

②如果 $y \in \Sigma_n$，令 $x = y$，转(2).

非形式地，I' 是这样一个产生式集合，它能导出一个相应的图 G'，而 $First(I')$，$Last(I')$ 则描述了图 G' 的嵌入关系，(它的作用如产生式规则 P 的 B_{in}, B_{out}是用来描述 $rhs(p)$的嵌入关系). $Before(S_k, P_k, I')$则描述了图 G' 中 S_k 的父结点的情况，$Fellow(S_k, P_k, I')$则描述了在图 G' 中 S_k 的子结点的情况.

记图 G 是被分析的一个图，$G = (N_G, E_G, \varphi_G)$，它有 n 个结点组成，$\varphi_G(n_1) = a_1, \cdots, \varphi_G(n_n) = a_n$，且 $a_1, \cdots, a_n$ 均为终极结点符，记 $a_1, \cdots, a_n$ 为文法分析时依次分析的结点符号，且 a_j 至少与 $a_1, \cdots, a_{j-1}$之一相连.

记 $I_1, \cdots, I_n$ 是相应于 $a_1, \cdots, a_n$ 的图文法分析表，且 $I_1, \cdots, I_n$ 是产生式集合的集合.

算法 3.1. 图文法分析算法

1. 构造 I_1：

(1)读入结点标号为 a_1 的结点.

(2)对所有 $P_1 \in R$，如果 $a_1 \in rhs(P_i)$，则 $\{P_i\} \in I_1$，且 P_i 中相应的标号为 a 的结点被标记为 $Used$.

(3)如果 $I_1 = \Phi$，则图 G 不是文法的句子，结束.

2. 从 I_{j-1}构造 I_j.

(1)读入标号为 a_j 的结点，a_j 至少与 $a, \cdots, a_{j-1}$之一相连.

(2)对每一个 $I' \in I_{j-1}$，I' 是产生式规则集合.

①存在产生式 $P_i \in I'$，如果 $a_j \in rhs(P_i)$，且 a_j 没有被标记为 $Used$，若 a_j 与 $a_1, \cdots, a_{j-1}$的关系与在图 G 中 a_j 与 $a_1, \cdots, a_{j-1}$的关系一致，则 $I' \in I_j$，且 P_i 中的 a_j 被标记.

②存在产生式 $P_k \in I'$，且 P_k 没有被标记为 $Used$，$S_k = lhs(P_k)$，存在 $P_i \in R$，$a_j, S_k \in rhs(P_i)$，如果满足：

(Ⅰ)在图 G 中有 $a_j \rightarrow a_{i_1}, \cdots, a_j \rightarrow a_{i_k}$，且 $\{a_{i_1}, \cdots, a_{ik}\} \subseteq \{a_1, \cdots, a_{j-1}\}$，如果在 P_i 中有 $(a_j \rightarrow S_k) \in rhs(P_i)$，且 $\{a_{i_1}, \cdots, a_{i_k}\} = First(I')$.

(Ⅱ)在图 G 中有 $a_j \leftarrow a_{k_1}, \cdots, a_j \leftarrow a_{i_k}$，且 $\{a_{k_1}, \cdots, a_{k_m}\} \subseteq \{a_1, \cdots, a_{j-1}\}$，如果在 P_i 中有 $(a_j \leftarrow S_k) \in rhs(P_i)$，且 $\{a_{k_1}, \cdots, a_{k_m}\} = Last(I')$.

则$\{P_i\}\cup I'\in I_j$,且P_k被标记为$Used$,P_i中的a_j与S_k也相应地被标记为$Used$.

③存在产生式$P_k\in I'$,且P_k没有被标记为$Used$,$S_k=lhs(P_k)$,存在P_i,存在$P_i,P_j\in\hat{R}$,$a_j\in rhs(P_i)$,$S_i=lhs(P_i)$,$S_i,S_k\in rhs(P_j)$,满足:

(Ⅰ)在图G中有$a_j\rightarrow a_{i_1},\cdots,a_j\rightarrow a_{i_k}$,且$\{a_{i_1},\cdots,a_{i_k}\}\subseteq\{a_1,\cdots,a_{j-1}\}$,如果在$P_i$中有$(S_i\rightarrow S_k)\in rhs(P_i)$,且$a_j\in B_{out}(P_i)$,$\{a_{i_1},\cdots,a_{i_k}\}=First(I')$.

(Ⅱ)在图G中有$a_j\leftarrow a_{k_1},\cdots,a_j\leftarrow a_{k_m}$,且$\{a_{k_1},\cdots,a_{k_m}\}\subseteq\{a_1,\cdots,a_{j-1}\}$,如果在$P_i$中有$(S_i\leftarrow S_k)\in rhs(P_i)$,且$a_j\in B_{out}(P_i)$,$\{a_{k_1},\cdots,a_{k_m}\}=Last(I')$.

则$\{P_i,P_j\}\cup I'\in I_j$,且$P_i$中的$a_j$与$P_j$中的$S_i,S_k$均被标记为$Used$,且$P_i,P_k$也被标记为$Used$.

④存在产生式$P_j\in I'$,$S_i\in rhs(P_j)$,且S_i没有被标记为$Used$,存在$P_i\in R$,$S_i=lhs(P_i)$,$a_j\in rhs(P_i)$,如果满足:

(Ⅰ)在图G中有$a_j\rightarrow a_{i_1},\cdots,a_j\rightarrow a_{i_k}$,且$\{a_{i_1},\cdots,a_{i_k}\}\subseteq\{a_1,\cdots,a_{j-1}\}$,如果$a_j\in B_{out}(P_i)$,$\{a_{i_1},\cdots,a_{i_k}\}=Fellow(S_i,P_j,I')$.

(Ⅱ)在图G中有$a_j\leftarrow a_{k_1},\cdots,a_j\leftarrow a_{k_m}$,且$\{a_{k_1},\cdots,a_{k_m}\}\subseteq\{a_1,\cdots,a_{j-1}\}$,如果$a_j\in B_{in}(P_i)$,$\{a_{k_1},\cdots,a_{k_m}\}=Before(S_i,P_j,I')$.

则$\{P_i\}\cup I'\in I_j$,且在P_i中的a_j与P_j中的S均被标记为$Used$,同时P_i也被标记为$Used$.

(3)如果$I_j=\Phi$,则图G不是文法的句子结束.

引理 3.1. 算法4.1中每个结点符号a_j的相应的分析表I_j中的每个元素$I'\in I_j$:满足:

(1)I'中有且仅有一个产生式规则没有被标记为$Used$,I'中任一产生式规则P_i,若P_i的右部有标记为$Used$的结点标号x,若$x\in\Sigma_t$,则$x\in\{a_1,\cdots,a_j\}$,若$x\in\Sigma_n$,则存在另一被标记为$Used$的产生式$P_j\in I'$,且$x=lhs(P_j)$.

(2)I'是这样一组产生式规则集合,它能产生这样的图型G_j,G_j中包含$a_1,\cdots,a_j$,且在图G_j中$a_1,\cdots,a_j$之间的相互联结关系与在图G中$a_1,\cdots,a_j$的相互联结关系一致.

证明:(1)由算法3.1对I_j的构造过程知,命题成立.

(2)对j作归纳证明.因为I_j中的元素I是由I_{j-1}中的元素I''所构造而成的.且I'的构造原则是a_j与$a_1,\cdots,a_{j-1}$的关系与在图G中a_j与$a_1,\cdots,a_{j-1}$的联结关系一致.由归纳假设,在I''中$a_1,\cdots,a_{j-1}$之间的联结关系与图G中$a_1,\cdots,a_{j-1}$的关系一致.因此在I'中$a_1,\cdots,a_j$之间的联结关系与在G中$a_1,\cdots,a_j$的联结关系一致.

引理 3.2. 分析表I_j中每个元素$I'\in I_j$,I'中的产生式规则可构造成一棵文法推导树.

证明:(1)设P_i是I'中唯一没有被标记的产生式,把P_i看成文法推导树的根.

(2)P_j是树中的任一结点,P_i中每个被标记的非终极结点符x,则相应地在I'中存在产生式P_k,$x=lhs(P_k)$,则P_k作为P_j的子结点.

(3)重复过程(2),直至I'中所有产生式均在树中.

定理 3.3. $G\in L(GG)$,当且仅当$I_n\neq\Phi$,I_n中至少存在一个元素I',I'中的产生式规则满足:

(1)P_i是I'中唯一没有标记的产生式,且$S_0=lhs(P_i)$.

(2)对$\forall P_j\in I'$,P_j的右部的所有结点均标记为$Used$.

证明：(⇒)因为 $G\in L(GG)$，存在一产生式序列 τ，$S_0\overset{\tau}{\Rightarrow}G$，对 τ 的产生式个数作归纳证明：

设 $S_0\overset{P}{\Rightarrow}H\overset{*}{\Rightarrow}G$，在图 H 中，有 k 个终极结点符，u 个非终极结点符 $S_1,\cdots,S_u$，且存在图 $G_1,\cdots,G_u$ 有 $S_1\overset{*}{\Rightarrow}G_1,\cdots,S_u\overset{*}{\Rightarrow}G_u$，$G=H(S_1\leftarrow G_1,\cdots,S_u\leftarrow G_u)$.

在读入结点 a_j 时，如果满足下列条件之一；

(1)$a_j\in H$，且 H 中其它任何结点还没有被读入：

①如果 $G_1,\cdots,G_u$ 中还没有任何终极结点符被读入，由算法 3.1(1)知，$P\in I'$，且 a_j 被标记为 *Used*.

②如果 $G_1,\cdots,G_u$ 中有唯一的图 G_k，G_k 中有结点被读入，设 I'' 是上次分析表中的一个元素，设 P_k 是 I'' 唯一未被标记的产生式，$S_k=lhs(P_k)$，由算法 3.2(2)(2)(2)知，$\{P\}\cup I''=I'$，且 P 中的 a_j 与 S_k 及 P_k 标记为 *Used*.

③如果 $G_1,\cdots,G_u$ 已有多个图中的结点被读入，则由算法 3.1(2)(2)(3)，在 I_{j-1} 中一定存在这样的一个元素 I''，$P\in I''$，由算法 3.1(2)(2)(1)知，$I'=I'$，且 P 中的 a_j 被标记.

(2)$a_j\in H$，且 H 中已有终极结点符被读入，则一定存在 $I''\in I_{j-1}$，且 $P\in I''$，由算法 3.1(2)(2)(1)知，$I''=I'$，且 P 中的 a_j 被标记.

(3)$a_j\in G_k$，且 G_k 中还没有结点被读入：

①如果 $G_{k'}$，$k'=1,\cdots,u$，$k'\neq k$，中还没有结点被读入，由归纳假设，可得分析表元素 I''，若 H 中还没有结点被读入，则 $I''=I'$. 若 H 中已有结点被读入，则 $\{P\}\cup I''=I'$，且 P 中的 S_k 被标记为 *Used*.

②如果存在唯一的 G'_k 已有结点被读入，$I''\in I_{j-1}$，由算法 3.1(2)(2)(3)，有 $P\in I'$，且 S_k，$S_{k'}$ 被标记为 *Used*.

③如果存在多个 G'_k 有结点被读入，存在 $I''\in I_{j-1}$，$P\in I''$(由上面的证明)，由算法 3.1(2)(2)(4)，P 中的 S_k 被标记为 *Used*.

(4)$a_j\in G_k$，且 G_k 中已有结点被读入，由归纳假设，可以得到相应的分析表.

记 $I_1',\cdots,I_u'$ 分别是 $S_1\overset{*}{\Rightarrow}G_1,\cdots,S_u\overset{*}{\Rightarrow}G_u$ 是分析表中满足相应条件的元素，则由算法 3.1，在 I_n 中存在元素 $I'=\{P\}\cup I_1'\cup I_2'\cup\cdots\cup I_u'$ 中未被标记的产生式在 I' 中也被标记，且 $S_0=lhs(P)$，P 是唯一没有被标记的产生式.

(⇐)由引理 3.2，命题成立.

记 K_1 是产生式规则集 R 中产生式规则的个数，K_2 是 R 中产生式右部的图的最大结点个数，记 λ_j 是 I_j 中元素个数，α_j 是 I_j 中所有元素的最大产生式个数.

由 $First(I')$，$Last(I')$，$Fellow(S_k,P_k,I')$，$Before(S_k,P_k,I')$ 的定义，它们的时间复杂性为 $O(K_2\alpha_j)$.

算法 3.1(2)(2)(1)的时间复杂性为 $2*j_2*K_2*\alpha_j$.

算法 3.1(2)(2)(2)的时间复杂性为 $2*j_1*K_1*\alpha_j*K_2*K_2*\alpha_j$.

算法 3.1(2)(2)(3)的时间复杂性为 $2*j_1*K_1*\alpha_j*K_2*K_2*\alpha_j$.

算法 3.1(2)(2)(4)的时间复杂性为 $2*j_1*K_1*\alpha_j*K_2*K_2*\alpha_j$.

由 I' 的构造知：$\alpha_j\leqslant K_1*j$.

算法 3.1 的时间复杂性为 $\sum_{j=1}^{n}\sum_{i=1}^{\lambda_j}(2j\alpha_jK_2(3K_1\alpha_jK_2+1))$. 如果 λ_j 是有界的,记为 M,由算法 3.1 的时间复杂性为 $O(n_4\cdot M)$,即为多项式时间的.

但在最坏情况下,$\lambda_j=k_1^j$,则算法 3.1 是指数阶的.

记 G_j 是 G 的一个子图,且 G_j 只包含且仅包含 $a_1,\cdots,a_j$ 的图,由引理 3.2,I_j 中的每个元素都能导出包含图 G_j 的图. 一般情况下,能导出包含图 G_j 的产生式规则集是有限的,因此平均情况下,λ_j 是有界的,即平均情况下,算法是多项式时间复杂性的.

该算法用 C 语言已经在 386 上实现,从目前所作的分析测试中,效果良好.

4　结束语

本文通过对上下文无关图文法的推导的分析,证明了推导对非终极结点符是独立的,并给出了一种有效的自上而上图文法分析算法,它具有多项式时间复杂性,这一技术将在句法模式识别、图象处理等方面具有积极的作用.

参考文献

1 Ehrig H *et al*. Graph—grammar and their application to computer science. Lecture Note in Computer Science, 1986.

2 Rozenberg G, Welzl E. Boundary NLC graph grammar—basic definition, normal forms and complexity. Information and Control, 1986,**69(1)**:136—167.

3 Lauteman C. The complexity of graph language generated by hyperedge replacement. Acta Information, 1990,**27(3)**:399—421.

4 Vigna P D, Gheezi C. Context—free graph grammar. Formation and Control, 1978,**33(2)**:207—233.

5 Frank F. A class of linearly passable graph grammar. Acta Information, 1983,**10(2)**:175—201.

6 Shi Q Y, Fu K S. Parsing and translation of (attributed) expansive graph larguage for scene analysis. IEEE PAMI—5 1983,**5(5)**:472—485.

7 花全香. 上下文无关图文法和自动程序理解系统[博士论文]. 东南大学,1993.

8 Earley J. An efficient context—free parsing algorithm. Communication of ACM, 1970,**13(2)**:94—102.

AN EFFICIENT PARSING ALGORITHM ON NODE LABEL CONTEXT FREE GRAPH GRAMMAR

Hua Quanxiang　Xing Hancheng　Feng Chunbo

(*Department of Computer Sciences, South East University, Nanjing* 210018)

Abstract　This paper discusses the property of context—free graph grammar firstly, and proves the order independence property of graph grammar derivation. An efficient graph parsing algorithm which has a polynomial time Complexity is proposed, and the correctness of the parsing algorithm is proved.

Key words　Context—free graph grammar, parsing.