含无序产生式的故事分析文法的研究*

张松懋

(中国科学院数学研究所,北京 100080)

摘要 基于故事分析文法的故事理解法是本文作者在其博士论文中提出的一种面向故事深层含义的对语法、语义和语用综合进行理解的形式化方法,故事分析文法是一种高维的上下文有关文法,为了提高它的描述能力和表达范围,本文定义了一种特殊的产生式:无序产生式,并将它引入故事分析文法,讨论了由此带来的对故事分析文法的影响和无序产生式在故事分析文法中的性质,最后给出了含无序产生式的故事分析文法的一个子类文法的语法分析算法.

关键词 人工智能,故事理解,高维文法,上下文有关文法,产生式,语法分析算法.

人工智能技术中,自然语言理解一直是一个比较困难的研究领域,故事理解是自然语言理解的重要分支之一,对它的研究开始于七十年代.由于缺乏形式化方法的支持或以往的形式化方法难以理解语用,故事理解的研究进展缓慢,仍处于一种探索阶段.鉴于这种情况,我们把描述分析计算机语言的形式化方法:形式语言和编译理论引入故事理解,在文[1]中提出了一种新的故事理解的形式化方法,用一种高维的上下文有关文法来表达故事的语法、语义和语用,构造这种文法的语法分析算法,并用它对故事进行分析识别从而理解故事的深层含义(如中心思想).这种故事理解称为故事分析,它所基于的文法称为故事分析文法,我们研究了故事分析文法的一个子类文法——弱优先故事分析文法的语法分析算法,并证明了该算法的正确性和无二义性.

本文从介绍[1]中的故事表达、故事分析文法、弱优先故事分析文法及其语法分析算法入手,讨论了故事分析文法在描述产生式中产生式树间非时间性排列方面的局限性,由此,为了提高故事分析文法的描述能力和表达范围,本文定义了一种特殊的产生式——无序产生式,并将它引入故事分析文法以解决产生式中产生式树间非时间性排列问题,讨论了由此带来的对故事分析文法的影响和无序产生式在故事分析文法中的性质,最后给出了含无序产生式的弱优先故事分析文法的语法分析算法.

1 关于[1]中的故事表达

1 正叙结构的故事

* 本文1993年4月23日收到,1993年6月15日定稿

作者张松懋,女,26岁,助研,1992年博士毕业于中科院数学所,主要研究领域为AI,故事理解,分布式系统.
本文通讯联系人:张松懋,北京 100080,中国科学院数学研究所

一般说来,一个故事由若干某种自然语言的句子组成,我们把这样的句子称为动作型句子:含有一个动词且表示一个动作,这个动作可以是主动者的形体动作,也可以是思维动作,还可以是某种状态.若一个故事只包含动作型句子,而且故事的叙述是按动作发生的时间先后次序进行的,则称该故事具有正叙结构.本文以后提到故事均指具有正叙结构的故事.

2　故事的格框架森林表达

我们先来说明一些符号和概念:

• 称符号的非空集合为字母表.

• 一个树 T 是含一个或一个以上元素(称为节点)的有穷集,而且存在一个唯一的指定为根的节点,而且其余节点分为 m 个从左至右互不相交的集合:$T_1,T_2,\cdots,T_m(m\geqslant 0)$,每个集合都是一棵树,其中每棵树的根称为树 T 的根的一个子节点,每个 $T_i(1\leqslant i\leqslant m)$中的每个节点都称为树 T 的根的后代节点.

• 若称树 T 中一个节点为节点 a,则表示符号 a 是该节点的一个标记.树 T 中,以 T 中除根以外的任一节点为根的包含该节点的所有后代节点的树称为 T 的子树.

• 设 V 是字母表,T_V 是所有这样的树的集合,其中树的节点均是以 V 中符号为标记的.设 $N\subset V,N\neq\varnothing$,则 $T_V^{[N]}\subset T_V$ 且 $T_V^{[N]}$ 中的每棵树上至少有一个节点是以 N 中符号为标记的.

• 对任意自然数 n 及任意的 $t_1,\cdots,t_n\in T_V$,把 $t_1,\cdots,t_n$ 并置在一起所构成的串 $t_1\cdots t_n$ 称为 T_V 上的一个森林;对任意 $1\leqslant i\leqslant j\leqslant n$,称 $t_i\cdots t_j$ 为森林 $t_1\cdots t_n$ 的子森林;当 $n=0$ 时森林 $t_1\cdots t_n$ 称为空森林,它不含任何树,记为 ε;记 $T_V^*=\{\alpha\mid\alpha$ 为 T_V 上一个森林$\}$;记 $T_V^+=\{\alpha\mid\alpha\in T_V^*$ 且 $\alpha\neq\varepsilon\}$;记 $|t_1\cdots t_n|=n$.

下面我们来说明故事的格框架森林表达.

自然语言理解中句子理解的主要技术之一是 Fillmore 于 1968 年提出的格框架模型,它在分析句子时以动词为中心,其它成分均看作是对动词(动作)的修饰,每一种修饰称为一个格.格框架森林是基于自然语言理解中句子理解的一种故事表达方法,具体说来是:把故事中的每个句子分别用一棵格框架树表示,格框架树是树,按故事的叙述顺序把这些格框架树并置在一起,形成的森林称为格框架森林.可见格框架森林中格框架树的排列次序和它们所表达的动作的发生时间的先后次序是一致的.

2　关于[1]中的故事分析文法、弱优先故事分析文法及其语法分析算法

从故事的表达来看,故事分析文法不是一种简单的串文法,它的产生式应具有高维结构,同时它也不是上下文无关的文法,为此我们定义了一种新的高维文法——森林文法,故事分析文法是它的一个子类.下面我们以非形式化的方式来说明森林文法、故事分析文法、弱优先故事分析文法及其语法分析算法,以及与它们有关的和本文以后将用到的符号和概念,其中标以'△'的算法或概念的完整形式化定义请见[1].

△森林文法是一个四元组 $G_f=(N,\Sigma,P,S)$,令 $V=N\cup\Sigma$,其中,N 和 Σ 分别是非终结符集和终结符集,P 是产生式集,产生式的左部是不为空的森林,而且其中至少有一棵树属于 $T_V^{[N]}$,产生式的右部是可为空的森林,$S\in T_V^{[N]}$,称为起始树.

△设森林文法 $G_f=(N,\Sigma,P,S)$，$V=N\cup\Sigma$，$\alpha,\beta\in T_V^*$，称 α 一步推导出 β，记为 $\alpha\Rightarrow\beta$，当且仅当 α 串推导出 β 或 α 树推导出 β，其含义分别是根据某产生式来替换 α 的一个子森林或替换 α 中的一棵树的一棵子树从而得到 β. 令 $\alpha_1,\alpha_2,\cdots,\alpha_n(n\geqslant 1)\in T_V^*$，若 $\alpha_1\Rightarrow\alpha_2\Rightarrow\cdots\Rightarrow\alpha_n$，则称这个序列是从 α_1 到 α_n 的一个推导，若存在从 α_1 到 α_n 的推导，则称 α_1 可推导出 α_n；记 $\alpha_1\overset{+}{\Rightarrow}\alpha_n$ 表示从 α_1 出发经一步或若干步可推导出 α_n；记 $\alpha_1\overset{*}{\Rightarrow}\alpha_n$ 表示 $\alpha_1=\alpha_n$ 或 $\alpha_1\overset{+}{\Rightarrow}\alpha_n$；若 $\alpha_1=\alpha_n$ 则表示从 α_1 出发经 0 步可推导出 α_n；记 $\alpha_1\overset{m}{\Rightarrow}\alpha_n$ 表示从 α_1 出发经 m 步($m\geqslant 0$)可推导出 α_n.

• 设森林文法 $G_f=(N,\Sigma,P,S)$，$V=N\cup\Sigma$，$\alpha\in T_V^*$，$\omega\in T_\Sigma^*$，若 $S\overset{*}{\Rightarrow}\alpha$，则称 α 是 G_f 的一个树句型，若 $S\overset{*}{\Rightarrow}\omega$，则称 ω 是 G_f 产生的一个树句子.

• 设森林文法 G_f，若 G_f 的每个右部不为空的产生式 $\alpha\rightarrow\beta$ 满足 $|\alpha|\leqslant|\beta|$，则称 G_f 为上下文有关森林文法，若 G_f 还满足它的每个产生式 $\alpha_1\rightarrow\beta_1$ 都有 $|\alpha_1|=1$，则称 G_f 为单左上下文有关森林文法，记为 slcsfg.

• 设 slcsfg G_f，称 G_f 的产生式树集为由 G_f 中所有产生式中的左部树和右部森林中的所有树组成的集合，记为 $PRT(G_f)$，$PRT(G_f)$ 中的元素称为 G_f 的一棵产生式树；称 G_f 的非算符树集为由 G_f 中所有产生式的左部树组成的集合，记为 $NOTS(G_f)$，其中的元素称为 G_f 的一个非算符树. 显然有：$NOTS(G_f)\subset PRT(G_f)$.

• 设 slcsfg G_f，令 $t\in PRT(G_f)$，称 t 的树推导集为由 t 出发经 0 步或若干步树推导所能推出的所有的树的集合，记为 DERIV(t).

△设 slcsfg G_f，称产生式树 t 为树产生式树当且仅当 t 不是任何树句型中的树，即 t 只可能是树句型中树的子树，记 $TPRT(G_f)$ 为 $PRT(G_f)$ 中所有树产生式树的集合；若 G_f 的产生式 $t\rightarrow u$ 满足 $t\in TPRT(G_f)$ 且 $u\in PRT(G_f)$，则称它为树产生式.

• 设 slcsfg $G_f=(N,\Sigma,P,S)$，$V=N\cup\Sigma$，令 $t,u\in PRT(G_f)$，$T,U,U_1\in T_V^{[N]}$，$\alpha_1,\alpha_2,\beta_1,\beta_2\in T_V^*$，产生式树偶(t,u)存在优先关系，当且仅当下列三种情况之一存在：(i)$t\doteq u$，当且仅当 P 中有产生式 $T\rightarrow\alpha_1 t u\alpha_2$；(ii)$t\lessdot u$，当且仅当 P 中有产生式 $T\rightarrow\alpha_1 tU\alpha_2$，且 $U\overset{+}{\Rightarrow}u\beta_i$ 且其中每步都是串推导；(iii)$t\gtrdot u$，当且仅当 P 中有产生式 $T\rightarrow\alpha_1 UU_1\alpha_2$，而且 $U\overset{+}{\Rightarrow}\beta_1 t$ 且 $U_1\overset{*}{\Rightarrow}u\beta_2$，且这两个推导中的每一步都是串推导.

• 设 slcsfg G_f，对任意 $t,u\in PRT(G_f)$，$t\lessdot\doteq u$ 当且仅当 $t\doteq u$ 或 $t\lessdot u$.

△一般的森林文法并不适合作故事分析之用，为此，作为单左上下文有关森林文法的一个子类，根据故事的格框架森林表达，我们定义了故事分析文法，利用故事分析文法的产生式进行推导的过程就是对故事内容从抽象到具体的逐步生成过程，针对故事内容具有很强的上下文有关性的特点，我们在故事分析文法中定义了一种上下文有关推导.

△作为故事分析文法的一个子类，我们定义了弱优先故事分析文法. 在弱优先故事分析文法中，对任意产生式树 t,u，若它们都不是树产生式树，则树偶(t,u)至多只满足一种如下的优先关系：$t\lessdot\doteq u$，$t\gtrdot u$；空产生式都是树产生式；若两个产生式的右部相同，则它们都是树产生式；对任意产生式树 t 和 u，若 $t\neq u$ 而且它们的树推导集的交集不为空，则 t 和 u 至少有一个是树产生式树.

△设弱优先故事分析文法 $G_s=(N,\Sigma,P,S)$，$V=N\cup\Sigma$，G_s 的语法分析算法的功能是判

断一个给定的格框架森林是否是 G_s 的树句子,若是,则表示该格框架森林(代表一个故事)可以被抽象成 G_s 中起始树所代表的分析目标.

下面我们来介绍 G_s 的语法分析算法中所用到的 G_s 的优先关系表,它存贮着 G_s 的产生式树之间的优先关系,是一个二维矩阵 pret(T,U),其中 $T,U\in PRT(G_s)\cup\{\#\}$ 且 $T,U\notin TPRT(G_s)$,'#'是一个特殊符号,$\#\notin V$,'#'是输入森林的结束符号,把它也看作是 G_s 的一个产生式树,而且规定:对任意 $t\in PRT(G_s)$ 且 $t\notin TPRT(G_s)$,有:$t\gtrdot\#$. 对任意 $T,U\in PRT(G_s)\cup\{\#\}$ 且 $T,U\notin TPRT(G_s)$,pret(T,U)的值共有三种可能:$\lessdot$,$=$,$\gtrdot$和空白符,空白符表示 T 和 U 之间没有优先关系,因 G_s 是弱优先故事分析文法,所以可知 pret(T,U)的值有且仅有一个.

G_s 的语法分析算法中有一个匹配子函数,它的功能是判断一个森林 $t_1\cdots t_n$ 是否和一产生式 $T\to u_1\cdots u_m$ 匹配成功,其中 $t_1,\cdots,t_n,T,u_1,\cdots,u_m\in T_V$,匹配成功是指:$n=m$ 且对任意 $i,1\leqslant i\leqslant n$,有:$t_i=u_i$ 或 $t_i\in DERIV(u_i)$.

3　故事分析文法的产生式中产生式树的非时间性排列和无序产生式

我们是针对具有正叙结构的故事构造故事分析文法的,因此,在故事分析文法中,产生式右部森林中产生式树的排列次序就表示了这些树所代表的语义含义的时间先后次序,这意味着产生式中产生式树的排列是一种时间性排列.但是,常常某些产生式树所代表的含义之间不存在时间先后次序的要求,例如:

"若 X 喜欢打球(用树 T_1 表示),X 还喜欢下棋(用 T_2 表示),X 还喜欢跳舞(用树 T_3 表示),则可以得出 X 爱玩(用 T 表示)的结论."

其中 T_1,T_2,T_3 之间没有时间先后次序的要求,为表达这个信息的含义,需要六条产生式:

$T\to T_1T_2T_3$　　$T\to T_1T_3T_2$　　$T\to T_2T_1T_3$

$T\to T_2T_3T_1$　　$T\to T_3T_1T_2$　　$T\to T_3T_2T_1$

当产生式右部森林中有 $n(n\geqslant 2)$ 个产生式树之间都不存在时间性要求时,就需要 $P_n^n=n!$ 个产生式,这会使故事分析文法的产生式集合非常庞大从而存贮空间大且处理效率非常低,这表明故事分析文法在描述产生式中产生式树的非时间性排列方面有局限性,为此,我们引入一种特殊的产生式——无序产生式:

定义 1. 设森林文法 $G_f=(N,\Sigma,P,S)$,$V=N\cup\Sigma$,$@\notin V$,称具有下列形式的产生式为无序产生式:

$$T\to\beta_1@\alpha@\beta_2$$

其中,$T\in T_V^{[N]}$,$\beta_1,\beta_2\in T_V^*$,$\alpha\in T_V^+$ 且 $|\alpha|\geqslant 2$,令 $n=|\alpha|$,这个无序产生式代表了 n! 个产生式,这 n! 个产生式分别为:$T\to\beta_1\alpha_1\beta_2,\cdots,T\to\beta_1\alpha_k\beta_2$,其中 $k=n!$,$\alpha_1,\cdots,\alpha_k$ 是通过对 α 中的 n 棵树进行全排列而得到的 n! 个森林. #

我们在故事分析文法中引入无序产生式,以解决产生式中产生式树的非时间性排列问题,从而使得在不增加产生式的前提下提高了故事分析文法的表达能力和描述范围.

在故事分析文法中引入无序产生式,从故事分析文法的推导来看,因为无序产生式的右部森林长度大于 1,它不是树产生式,所以它不会对树推导产生影响;无序产生式代表了一

组具有相同左部树的产生式，使用无序产生式的串推导实际上是使用它所代表的一组产生式中的一个产生式进行推导，因此无序产生式对串推导也不会发生影响.

从故事分析文法的分析过程来看，特别地，在弱优先故事分析文法的分析算法中，优先关系表的构造算法和匹配子函数都要修改以针对无序产生式进行特殊处理.

根据无序产生式的定义和单左上下文有关森林文法中产生式树偶之间的优先关系的定义，可以得出无序产生式在故事分析文法中的性质如定理1所示.

定理1. 设故事分析文法 $G_s=(N,\Sigma,P,S)$，$V=N\cup\Sigma$，对任意无序产生式 $T\to t_1\cdots t_l@u_1\cdots u_m@t_{l+1}\cdots t_n$，$0\leqslant l\leqslant n$，$m\geqslant 2$，$t_1,\cdots,t_n,u_1,\cdots,u_m\in PRT(G_s)$，下列几种情况均成立：

(i)对任意 t_i,t_{i+1}，$1\leqslant i\leqslant l-1$ 或 $l+1\leqslant i\leqslant n-1$，下列三个式子均成立：

(i−1)$t_i\asymp t_{i+1}$.

(i−2)对任意 $u\in PRT(G_s)$，若存在 $\alpha\in T_V^*$ 使得有 $t_{i+1}\Rightarrow u\alpha$ 且其中每步都是串推导，则有：$t_i\lessdot u$.

(i−3)对任意 $t,u\in PRT(G_s)$，若存在 $\alpha,\beta\in T_V^*$，使得有 $t_i\overset{+}{\Rightarrow}\beta t$ 且其中每步都是串推导，和 $t_{i+1}\overset{*}{\Rightarrow}u\alpha$ 且其中每步都是串推导，则有：$t\gtrdot u$.

(ii)对任意 u_i，$1\leqslant i\leqslant m$，下列式子均成立：

(ii−1)$t_l\asymp u_i$.

(ii−2)$u_i\asymp t_{l+1}$.

(ii−3)对任意 u_j，$j\neq i$ 且 $1\leqslant j\leqslant m$，下列式子均成立：

(ii−3−1)$u_i\asymp u_j$.

(ii−3−2)对任意 $u\in PRT(G_s)$，若存在 $\alpha\in T_V^*$ 使得有 $u_j\overset{+}{\Rightarrow}u\alpha$ 且其中每步都是串推导，则有：$u_i\lessdot u$.

(ii−3−3)对任意 $t,u\in PRT(G_s)$，若存在 $\alpha,\beta\in T_V^*$ 使得有 $u_i\overset{+}{\Rightarrow}\beta t$ 且其中每步都是串推导，和 $u_j\overset{*}{\Rightarrow}u\alpha$ 且其中每步都是串推导，则有：$t\gtrdot u$.

(ii−4)对任意 $u\in PRT(G_s)$，若存在 $\alpha\in T_V^*$ 使得有 $u_i\overset{+}{\Rightarrow}u\alpha$ 且其中每步都是串推导，则有：$t_l\lessdot u$.

(ii−5)对任意 $t,u\in PRT(G_s)$，若存在 $\alpha,\beta\in T_V^*$ 使得有 $t_l\overset{+}{\Rightarrow}\beta t$ 且其中每步都是串推导，和 $u_i\overset{+}{\Rightarrow}u\alpha$ 且其中每步都是串推导，则有 $t\gtrdot u$.

(ii−6)对任意 $u\in PRT(G_s)$，若存在 $\alpha\in T_V^*$ 使得有 $t_{l+1}\overset{+}{\Rightarrow}u\alpha$ 且其中每步都是串推导，则有：$u_i\lessdot u$.

(ii−7)对任意 $t,u\in PRT(G_s)$，若存在 $\alpha,\beta\in T_V^*$ 使得有 $u_i\overset{+}{\Rightarrow}\beta t$ 且其中每步都是串推导，和 $t_{l+1}\overset{*}{\Rightarrow}u\alpha$ 且其中每步都是串推导，则有：$t\gtrdot u$. #

4　含无序产生式的弱优先故事分析文法的语法分析算法

设弱优先故事分析文法 $G_s=(N,\Sigma,P,S)$,$V=N\cup\Sigma$,含无序产生式的 G_s 的语法分析算法中优先关系表的构造算法和匹配子函数不同于不含无序产生式的 G_s 的语法分析算法,令 G_s 中含有无序产生式,下面给出针对无序产生式的 G_s 的优先关系表的构造算法和匹配子函数,不是无序产生式的处理和 G_s 的语法分析算法中其它部分这里不再赘述,不妨假定 G_s 中的产生式都是无序产生式.

1　优先关系表的构造算法

对每个 $U\in NOTS(G_s)$,定义 U 的两个集合 FIRSTPT+(U)和 LASTPT+(U)如下:

$FIRSTPT+(U)=\{t\mid U\overset{+}{\Rightarrow}t\alpha$,其中每步都是串推导且 $t\in PRT(G_s),\alpha\in T_v^*\}$

$LASTPT+(U)=\{t\mid U\overset{+}{\Rightarrow}\alpha t$,其中每步都是串推导且 $t\in PRT(G_s),\alpha\in T_v^*\}$

首先求每个非算符树 U 的 FIRSTPT+(U)集合:

```
AFIRSTPTT(Gs)
begin
    for 每个 U∈NOTS(Gs)和每个 t∈PRT(Gs)
        F(U,t):=false;
    for 每个形如 U→β1@α@β2 的产生式(|α|≥2,β1,β2∈T∇)
    begin
        if β1=ε then for α 中每个树 t
                    if F(U,t)=false then begin F(U,t):=true;
                                               把(U,t)下推进栈 STACK;
                                         end;
        if β1≠ε then begin
                    令 β1=tβ1',t∈Tv,β1'∈T∇;
                    if F(U,t)=false then begin F(U,t):=true;
                                               把(U,t)下推进栈 STACK;
                                         end;
                    end;
        end
while STACK 非空 do
begin
    推出 STACK 栈项(T,t);
    for 每个形如 U→@α@β 的产生式(|α|≥2,β∈T∇)
        if α 中存在树 T 且 F(U,t)=false
        then begin F(U,t):=true;
                   把(U,t)下推进栈 STACK;
             end;
    for 每个形如 U→Tβ1@α@β2 的产生式(|α|≥2,β1,β2∈Tv*)
        if F(U,t)=false then begin F(U,t):=true;
                                   把(U,t)下推进栈 STACK;
                             end;
end;
end
```

由此,对每个 $U\in NOTS(G_s)$,有:FIRSTPT+(U)={t|F(U,t)=true}. LASTPT+(U)的构造方法与此类似,不再赘述. 下面是 G_s 的优先关系表 pret 的构造算法:

```
APRETT(Gs)
```

```
begin
    for 每个产生式 T→t1…tl@u1…um@tl+1…tn
        (0≤l≤n,m≥2,T,t1,…,tn,u1,…,um∈TV)
    begin
        for k:=1 to 2
        begin
          k1:=1;
          k2:=l-1;
          for i:=k1 to k2
          begin pret(ti,ti+1):='≪≍';
                if ti+1∈NOTS(Gs)
                then for 每个 u∈FIRSTPT+(ti+1)
                          pret(ti,u):='≪≍';
                if ti∈NOTS(Gs)
                then for 每个 t∈LASTPT+(ti)
                          for 每个 u∈FIRSTPT+(ti+1)∪{ti+1}
                              pret(t,u):='≫';
          end;
          k1:=l+1;
          k2:=n-1;
        end;
        for i:=1 to m
        begin
          pret(ti,ui):='≪≍'
          pret(ui,ti+1):='≪≍';
          for j:=1 to m
            if i≠j
            then begin pret(ui,uj):='≪≍';
                       if uj∈NOTS(Gs)
                       then for 每个 u∈FIRSTPT+(uj)
                                 pret(ui,u):='≪≍';
                       if ui∈NOTS(Gs)
                       then for 每个 u∈LASTPT+(ui)
                                 for 每个 u'∈FIRSTPT+(uj)∪{uj}
                                     pret(u,u'):='≫';
                 end;
          if ul∈NOTS(Gs)
          then begin for 每个 u∈FIRSTPT+(ui)
                           pret(ti,u):='≪≍';
                     for 每个 u∈LASTPT+(ui)
                         for 每个 u'∈FIRSTPT+(tl+1)∪{tl+1}
                             pret(u,u'):='≫';
               end;
          if t1∈NOTS(Gs)
          then for 每个 t∈LASTPT+(t1)
                    每个 u∈FIRSTPT+(ui)∪{ui}
                        pret(t,u):='≫';
          if tl+1∈NOTS(Gs)
          then for 每个 u∈FIRSTPT+(tl+1)
                    pret(ui,tl+1):='≪≍';
    end;
end
```

2 匹配子函数

令：$\gamma=t_1\cdots t_n$，$n\geqslant 1$，$\beta_1=u_1\cdots u_l$，$\beta_2=u_{h+1}\cdots u_m$，$\alpha=u_{l+1}\cdots u_h$，$l\geqslant 0$，$2\leqslant h\leqslant n$，其中，$t_1,\cdots,t_n,u_1,\cdots,u_m,T\in T_V$，森林$\gamma$和无序产生式$T\to\beta_1@\alpha@\beta_2$，匹配成功当且仅当下列条件

均满足:

(i)$T \notin TPRT(G_s)$且对任意 u_i,$1 \leqslant i \leqslant m$,有:$u_i \notin TPRT(G_s)$.

(ii)$n=m$ 且对任意 t_i,$1 \leqslant i \leqslant l$ 或 $h+1 \leqslant i \leqslant n$,有:$t_i=u_i$ 或 $t_i \in DERIV(u_i)$.

(iii)对任意 t_i,$l+1 \leqslant i \leqslant h$,存在 u_j,$l+1 \leqslant j \leqslant h$,使得有:$t_i=u_j$ 或 $t_i \in DERIV(u_j)$.

(iv)对任意 u_j,$l+1 \leqslant j \leqslant h$,存在 t_i,$l+1 \leqslant i \leqslant h$,使得有:$u_j=t_i$ 或 $t_i \in DERIV(u_j)$.

单左上下文有关森林文法中,右部森林长度大于 1 的产生式中,左部树和右部森林中每个树都不是树产生式树.[1]无序产生式的右部森林长度大于 1,所以上面条件(i)恒满足.下面给出匹配子函数 match(γ,$T \rightarrow \beta_1 @ \alpha @ \beta_2$),匹配成功与否的返回值分别为'success'和'fail',其中,对任意 $t \in T_V$,$u \in PRT(G_s)$,判断是否 $t \in DERIV(u)$的算法请见[1].

```
match(γ, T→β1@α@β2)
begin
    if n≠m then return ('fail');
    for L1:=1 to 2
    begin
        K1:=1; K2:=1;
        for K:=K1 to K2
          if tK≠uK 且 tK∉DERIV(uK) then return ('fail');
        K1:=h+1;
        K2:=n;
    end;
    for K1:=l+1 to h
    begin
        L1:=0; L2:=0;
        for K2:=l+1 to h
        begin
          if tK1=uK2或 tK1∈DERIV(uK2)then L1:=1;
          if uK1=tK2或 tK2∈DERIV(uK1)then L2:=1;
        end;
        if L1=0 或 L2=0 then return('fail');
    end;
    return('success');
end
```

致谢　本文是在陆汝钤教授的悉心指导下完成的,谨致深深的谢意.

参考文献

1　张松懋.故事分析的研究.博士论文.中国科学院数学研究所,1992.

2　陈火旺,钱家骅,孙永强.程序设计语言编译原理.北京:国防工业出版社,1980.

3　Gries D. Compiler construction for digital computers. New York: John Wiley & Sons, Inc., 1971.

RESEARCH ON STORY PARSING GRAMMAR WITH UNORDERED PRODUCTIONS

Zhang Songmao

(*The Institute of Mathematics, The Chinese Academy of Sciences, Beijing* 100080)

Abstract Story Parsing Grammar (abbr. SPG) based story understanding approach, proposed by the author, is a formal method orienting the deep level meaning of the story to understand the syntax, semantics and pragmatics of the story. SPG is high—dimensional and context—sensitive. In order to strengthen SPG's descriptive ability and widen SPG's representative scope, a special kind of production, called unordered production, is defined and introduced into SPG. Concequentially, influences of unordered production on SPG are discussed. Moreover the characteristics of the unordered production in SPG are discussed. The syntactic parsing algorithm of a subclass of SPG with unordered productions is given at last.

Key words Artificial intelligence, story understanding, high—dimensional grammar, context—sensitive grammar, production, syntactic parsing algorithm.