

# 范畴语法

冯志伟

(语言文字应用研究所 北京 100010)

[摘要]当代计算语言学中出现了词汇主义的倾向,范畴语法是词汇主义的典型代表。本文介绍范畴语法的基本概念和方法,着重介绍句法类型演算,并通过具体的实例,说明了英语动词短语的句法类型演算研究的一些重要成果。

[关键词]范畴语法 句法类型 句法类型演算

[中图分类号] H087 [文献标识码] A [文章编号] 1003-5397(2001)03-0100-11

## Categorial Grammar: A review

Feng Zhiwei

**Abstract:** The lexicalism is a new trend in contemporary computational linguistics, and the categorial grammar is the typical grammar of lexicalism. In this paper, the author introduces the basic conception and method of categorial grammar, especially the syntactic type calculus. Some important results in the syntactic type calculus of English verb phrases are illustrated through concrete examples.

**Key words:** categorial grammar, syntactic type, syntactic type calculus.

范畴语法(categorial grammar)是由著名数理逻辑专家巴尔-希列尔(Yehoshua Bar-Hillel, 1915-1975)提出的。在1958年,数学家兰姆贝克(J. Lambek)在《句子结构的数学》(The mathematics of sentence structure, 载 American Mathematical Monthly, Vol. 65, p154-170)中,提出了句法类型演算的理论,根据这种理论,可以辨识一个符号串是不是语言中成立的句子。1959年,巴尔-希列尔在《自然语言结构的判定程序》(Decision procedure for structure in natural language,

[收稿日期] 2001-09-01

[作者简介] 冯志伟,云南省昆明市人,北京大学中文系与中国科技大学研究生院信息科学系研究生毕业,双硕士,70年代末80年代初在法国格勒诺布尔理科医科大学应用数学研究所(IMAG-GETA)留学,专攻机器翻译和计算语言学。曾任德国夫琅禾费研究院新信息技术与通信系统研究部(FhG)研究员,德国特里尔大学教授,德国康斯坦茨高技术大学国际术语学与应用语言学中心(CITaL)技术顾问,现为教育部语言文字应用研究所研究员,计算语言学博士生导师。

载 *logique et analyse*, 2 - e annee, No 5) 中, 进一步发展了句法类型演算的理论, 详细讨论了自然语言结构的判定程序。由于句法类型是一种范畴, 因此, 在 1960 年巴尔 - 希列尔等在《论范畴语法和短语结构语法》(On categorial and phrase structure grammars, 载 *Bull. Res. Council Israel*, Sec. F. 9, 1 - 16) 中, 把这种理论称为范畴语法 (categorial grammar)。1970 年蒙塔鸠 (R. Montague) 将范畴语法应用于自然语言的句法研究, 提出了泛语法 (universal grammar) 的理论, 他把句法范畴和语义范畴结合起来, 通过范畴语法建立自然语言到某个逻辑体系语义解释的对应关系, 从而进一步建立句法描述与语义描述之间的同构关系。范畴语法还成为类型逻辑语义学 (type logical semantics) 的主要部分。数十年来, 范畴语法一直是计算语言学研究关注的一个热点, 始终保持着其勃勃的生命力。早在 1975 年, 冯志伟就在《计算机应用与应用数学》杂志上介绍过这种语法, 可惜当时正值文革动乱, 没有引起我国语言学界和计算机界足够的重视; 由于近年来语言学中词汇主义日益盛行, 我国一些计算语言学学者又开始关注到范畴语法, 并试图应用范畴语法来建立汉语语义范畴的演算系统。但是, 许多从事计算语言学研究的专业人员对于范畴语法所知极少, 为了满足读者的要求, 本期我们比较细致地介绍这种语法。介绍时我们尽量避免过分形式化的叙述, 力图结合具体的语言实例来介绍。

我们首先介绍句法类型 (syntactic types) 的概念。

任何词都可以根据它在句子中的功能归入一定的句法类型, 如果用  $n$  表示名词的句法类型, 用  $S$  表示句子, 则其它的一些句法类型都可以用  $n$  和  $S$  以不同的方式结合起来表示。规则是:

如果有某个词  $B$ , 其后面的词  $C$  的句法类型是  $X$ , 而它们所构成的词的序列  $BC$  的功能与  $Y$  相同, 则这个词  $B$  的句法类型记为  $X/Y$ ;

如果有某个词  $B$ , 其前面的词  $A$  的句法类型为  $X$ , 而它们所构成的词的序列  $AB$  的功能与  $Y$  相同, 则这个词  $B$  的句法类型记为  $X \setminus Y$ ;

如果有某个词  $B$ , 其前面的词  $A$  的句法类型为  $X$ , 其后面的词的句法类型为  $Y$ , 而它们所构成的词的序列  $ABC$  的功能与  $Z$  相同, 则这个词  $B$  的句法类型为  $X \setminus Y / Z$ 。

根据这种记法, 可以写出自然语言中词的句法类型。

例如, 在英语中,

John 的句法类型为  $n$ 。

poor John (可怜的约翰) 中的 poor, 它后面出现名词 John, 而它所构成的 poor John, 功能与名词相同, 故其句法类型为  $n/n$ 。

John works (约翰工作) 中的 works, 它前面出现名词 John, 而它所构成的 John works, 功能与句子相同, 故其句法类型为  $n \setminus S$ 。

John likes Jane (约翰喜欢珍妮) 中的 likes, 它前面为名词 John, 后面为名词 Jane, 而它所构成的 John likes Jane 功能与句子相同, 故其句法类型为  $n \setminus S/n$ 。

John slept soundly (约翰睡熟了) 中的 soundly, 它前面的 slept 为  $n \setminus S$ , 而它所构成的 slept soundly, 功能与  $n \setminus S$  相同, 故其句法类型为  $(n \setminus S) \setminus n \setminus S$ 。

John works here (约翰在这里工作) 中的 here, 能够把 John works 这个句子  $S$ , 转换成一个新的句子 John works here, 故 here 的句法类型为  $S \setminus S$ 。

John never works (约翰从不工作) 中, 由于 John 的句法类型为  $n$ , 故 never works 的句法类型为  $n \setminus S$ , 可见, 句法类型为  $n \setminus S$  的 works 前面加了 never 构成 never works 之后, 其句法类型仍然

为  $n \setminus S$ , 所以, never 的句法类型为  $n \setminus S / (n \setminus S)$ .

John works for Jane (约翰为珍妮工作) 中, for 的作用与 John works here 中的 here 的作用相似, 但其后还有一个名词 Jane, 故 for 的句法类型为  $S \setminus S/n$ .

John works and Jane rests (约翰工作而珍妮休息) 中 and 是一个连接词, 它把前后两个句子连接起来, 构成一个新的句子, 故其句法类型为  $S \setminus S/S$ .

于是我们得到了如下的英语的句法类型表:

	词	句法类型	词类
(1)	John	$n$	名词
(2)	poor	$n/n$	形容词
(3)	works	$n \setminus S$	不及物动词
(4)	likes	$n \setminus S/n$	及物动词
(5)	soundly	$(n \setminus S) \setminus n \setminus S$	副词
(6)	here	$S \setminus S$	副词
(7)	never	$n \setminus S / (n \setminus S)$	副词
(8)	for	$S \setminus S/n$	介词
(9)	and	$S \setminus S/S$	连接词

从上表中可以看出, 句法类型大致相当于传统语法中的词类。(1) 中的 John 是名词, (2) 中的 poor 是形容词, (3) 中的 works 是不及物动词, (4) 中的 likes 是及物动词, (5) 中的 soundly, (6) 中的 here, (7) 中的 never 都是功能不尽相同的副词, (8) 中的 for 是介词, (9) 中的 and 是连接词。这样, 范畴语法便把英语的词类用  $S$  和  $n$  两个最基本的范畴表示出来。 $S$  和  $n$  是原子范畴, 而用它们表示出来的其他的词类则看成复合范畴。在逻辑语义的层面上,  $S$  代表了陈述句所表示的真值命题 (proposition),  $n$  代表了该命题中的论元 (argument), 这是一种非常简捷的句子的表示方法。在数学上, 如果我们把句子中除了  $S$  和  $n$  之外的语言单位都看成是函数, 把将和它们结合成新结构的那些语言单位看成该函数的变元, 那么, 函数的值便是两者合成所得到的那个新的结构。这样, 任何一个单词的语法特征便都可以通过这些原子范畴和复合范畴表示出来。这是一种典型的词汇主义 (lexicalism) 做法。可以说, 在计算语言学中, 范畴语法是词汇主义的典型代表。

对于语言中的词列出了一个完整的句法类型清单之后, 便可根据如下规则进行句法类型演算 (syntactic calculus):

如果有形如  $\_ , \setminus /$  的符号序列, 那么, 就用  $\_$  来替换它。

这个规则同时也包括下面两个规则:

用  $\_$  替换形如  $\_ , \setminus$  的符号序列, 即:  $( ) ( \setminus )$

用  $\_$  替换形如  $/$  的符号序列, 即:  $( / ) ( )$

如果把语言中的词标上句法类型, 通过有穷个演算步骤, 可把词的序列化为  $S$ , 则这个词的序列便是该语言中的合格句子。这样一来, 语言中各种成分的句法行为, 便都可以通过原子范畴和复合范畴的演算来描述了。

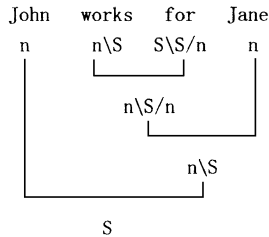




个句法类型不能使用我们前面讲过的演算规则合并,为此,兰姆贝克补充了如下的演算规则:

$$\begin{aligned} (\ \backslash ) (\ \backslash ) \quad \backslash \\ ( / ) ( / ) \quad / \end{aligned}$$

采用这样的规则,可对例5首先归并第2和第3个句法类型,作如下的演算:



这样一来,范畴语法中句法类型演算的演算规则就变成了如下4条:

$$\begin{aligned} ( ) (\ \backslash ) \\ ( / ) ( ) \\ (\ \backslash ) (\ \backslash ) \quad \backslash \\ ( / ) ( / ) \quad / \end{aligned}$$

明眼人不难看出,范畴语法在确定这些演算规则时显然考虑到了语义,不过,范畴语法的语义是通过句法类型以及反映这些句法类型的语义连锁的演算规则潜在地表示出来的。这种别具一格的表达方式,使得范畴语法的风格与短语结构语法的风格迥然不同,短语结构语法力图对句子进行切分,采用的是一种解析模式(analytic pattern),而范畴语法则力图反映句法类型的语义连锁,采用的是一种构造模式(constructive pattern)。范畴语法尽量设法把语义直接表示于句法之中,因而受到了计算语言学家的欢迎,其算术上的透彻性和模型的简明性,40多年来始终保持着其旺盛的生命力。

范畴语法对于英语动词短语的句法类型演算进行过细致的研究,这里简述这方面的内容。

英语动词短语的情况比较复杂,因此,在演算时,除了使用上述的  $n$  和  $S$  等句法类型符号之外,还应该再增加如下的句法类型符号:

- $i$ : 表示不及物动词的不定式
- $p$ : 表示不及物动词的现在分词
- $q$ : 表示不及物动词的过去分词

下面我们说明如何使用这些句法类型符号。

John must work (约翰必须工作)中,work 是不及物动词不定式,故其句法类型为  $i$ , must 前面为  $n$ ,后面为  $i$ ,它所构成的 John must work 的功能与句子相同,故其句法类型为  $n \backslash S/i$ 。

John is working (约翰正在工作)中,working 是不及物动词的现在分词,故其句法类型为  $p$ , is 前面为  $n$ ,后面为  $p$ ,它所构成的 John is working 的功能与句子相同,故其句法类型为  $n \backslash S/p$ 。

John has worked (约翰干了工作)中,worked 是不及物动词的过去分词,故其句法类型为  $q$ , has 前面为  $n$ ,后面为  $q$ ,它所构成的 John has worked 的功能与句子相同,故其句法类型为  $n \backslash S/q$ 。

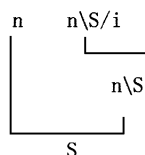
显而易见,我们之所以给 must, is, has 选择这样的句法类型,是为了保证 must work, is working, has worked 等的句法类型为  $n \backslash S$ ,使得它们象光杆动词 work 那样起作用。

John must be working (约翰一定在工作)中,working 是不及物动词的现在分词,故其句法类型为 p, be 后面为 working,它构成的 be working 的功能相当于一个不定式动词,故其句法类型为 i/p。

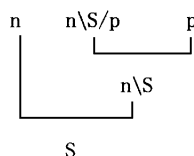
John has been working (约翰还在工作)中,working 是不及物动词的现在分词,故其句法类型为 p,been 后面为 working,它构成的 been working 的功能相当于一个不及物动词的过去分词,故其句法类型为 q/p。

我们举例来说明不及物动词句法类型演算的情况。

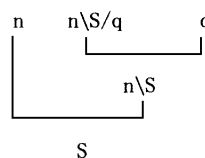
例 1. John must work



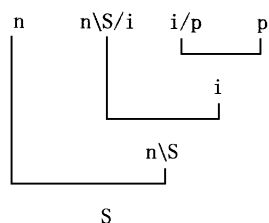
例 2. John is working



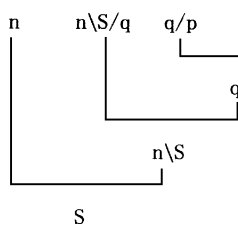
例 3. John has worked



例 4. John must be working



例 5. John has been working



我们再来看包含及物动词的句子情况。

John calls Jane (约翰给珍妮打电话)中,calls 的前面为 n,后面为 n,它所构成的 John calls Jane 的功能相当于一个句子 S,故 calls 的句法类型为 n \ S/n。

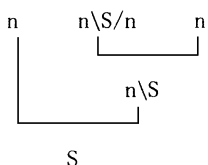
John must call Jane (约翰应该给珍妮打电话)中,call 的后面为 n,它所构成的 call Jane 的功能相当于一个不及物动词不定式 i,故其句法类型为 i/n。

John is calling Jane (约翰正在给珍妮打电话)中,calling 的后面为 n,它所构成的 calling Jane 的功能相当于一个不及物动词的现在分词 p,故其句法类型为 p/n。

John has called Jane (约翰给珍妮打了电话)中,called 的后面为 n,它所构成的 called Jane 的功能相当于一个不及物动词的过去分词 q,故其句法类型为 q/n。

我们举例来说明及物动词句法类型演算的情况。

例 6. John calls Jane

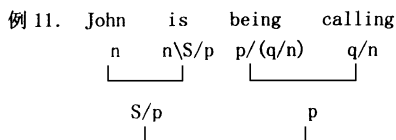


这里,calls 的句法类型与前面 likes 的句法类型是一样的,它们都是及物动词现在时单数第三





这是一个成立的句子。



这个句子的意思是：“有人正在给约翰打电话”，由句法类型演算的结果为 S 可以知道这是一个成立的句子。

上面的动词短语句法类型表是不完善的。表中没有列出一些重要的动词类型。例如，要求双宾语的动词，如 give(给)，appoint(任命)；把名词和形容词联系起来的动词，如“lunch tastes good”(中饭吃起来味道好)中的动词 tastes；像 am 这样的动词第一人称形式，像 are 这样的动词复数形式，等等。

表中列出的动词是有代表性的，它们实际上是某类动词的代表。例如，work 代表着所有的不及物动词，call 代表着所有的及物动词，must 代表着所有的情态助动词，如 will，shall，can，may，would，should，could，might 等。

表中的一些动词也可以有另外的句法类型。call 还可以是名词；have 和 be 还可以是实意动词，在句子“John must have lunch”(约翰必须吃午饭)以及句子“John must be good”(约翰必定是好的)中的 have 和 be 是实意动词。这时，它们就不能用表中的句法类型来表示了。

表中的第一列有空白，是因为 must 没有不定式，也没有 musting (p/i) 和 musted (q/i) 这样的形式。表中的第四、第五列中有空白，是因为助动词 have 没有现在分词，也没有过去分词，进行式助动词 be 没有现在分词。但是，实意动词 have，被动态助动词 be 以及实意动词 be 是有现在分词的。例如，

John is having lunch (约翰正在吃午饭)

John has had lunch (约翰吃了午饭)

John is being called (有人正在给约翰打电话)

其中，having 的句法类型是 p/n，had 的句法类型是 q/n，being 的句法类型是 p/(q/n)。

根据演算规则 - ，我们可以把表中的各个句法类型的演算结果总结为如下的乘法表(见下页)：

例如，当前项为 p/i，后项为 i 时，因为

$$(p/i) i \quad p$$

所以，p/i 与 i 相交之处为 p。

当前项为 q/i，后项为 i/p 时，因为

$$(q/i) (i/p) \quad q/p$$

所以，q/i 与 i/p 相交之处为 q/p。

乘法表中相交处的值可以反向展开。例如，由于乘法表中有 (p/i) i p，所以，可以把 p 反向展开为 (p/i) i；由于乘法表中有 (q/i) (i/p) q/p，因此，可以把 q/p 反向展开为 (q/i) (i/p)。

如果我们把乘法表中相交处的值都作这样的反向展开，那么，我们就可能对于语言现象获得一些新的认识。

例如，works 的句法类型为 n \ S，按乘法表把 n \ S 反向展开为 (n \ S/i) i，这意味着，works

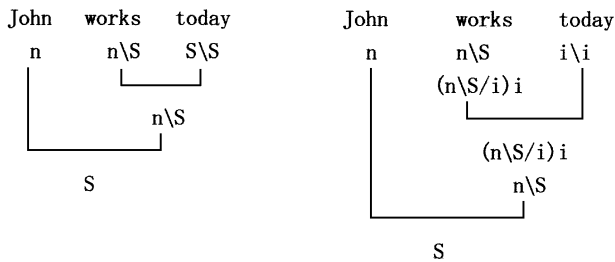
可以被看成是由句法类型分别为  $n \setminus S/i$  和  $i$  两部分组成的。在 *does work* 中, *does* 的句法类型为  $n \setminus S/i$ , *work* 的句法类型为  $i$ , 所以, 我们可以把 *does work* 解释为 *works* 的一种变体。事实上, 当 *works* 处于疑问句中的时候, 它就要变成 *does work*。

后项 \ 前项	$i/i$	$i$	$i/n$	$i/q$	$i/p$	$i/(q/n)$
$i/i$	$i/i$	$i$	$i/n$	$i/q$	$i/p$	$i/(q/n)$
$p/i$	$p/i$	$p$	$p/n$	$p/q$	$p/p$	$p/(q/n)$
$q/i$	$q/i$	$q$	$q/n$	$q/q$	$q/p$	$q/(q/n)$
$n \setminus S/i$	$n \setminus S/i$	$n \setminus S$	$n \setminus S/n$	$n \setminus S/q$	$n \setminus S/p$	$n \setminus S/(q/n)$

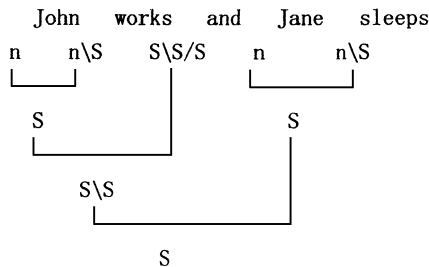
试比较, 陈述句: *John works* (约翰工作)

疑问句: *Does John work?* (约翰工作吗?)

又如, 副词 *today* 的句法类型为  $S/S$ , 在 *John works today* 中, 各个词的句法类型分别为  $n$ ,  $n \setminus S$  和  $S/S$ , 根据规则进行演算, 有:  $n(n \setminus S) \quad S, S(S/S) \quad S$ , 故这是一个成立句子。但是, 我们也可以把 *today* 的句法类型写为  $i/i$ , 因为如果我们把 *works* 的句法类型  $n \setminus S$  反向展开为  $(n \setminus S/i)i$ , 那么, 如果 *today* 的句法类型取  $i/i$ , 则有  $i(i/i) \quad i$ , 并且有  $(n \setminus S/i)i \quad n \setminus S$ , 因此,  $((n \setminus S/i)i)(i/i) \quad n \setminus S$ , 并且  $n(n \setminus S) \quad S$ 。这样, 也可以判定 *John works today* 是一个成立句子, 它的意思是“约翰今天工作”。这个句子的演算过程分别表示如下:

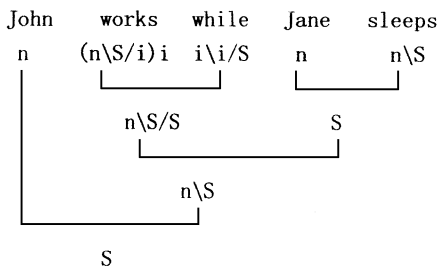


采用这种反向展开的办法, 还可以把等立连接词和从属连接词区别开来。例如, 在“*John works and Jane sleeps*”(约翰工作而珍妮睡觉)这个句子中, *and* 是等立连接词, 其句法类型是  $S \setminus S/S$ , 演算过程为:



但是, *works* 的句法类型  $n \setminus S$  也可以反向展开为  $(n \setminus S/i)i$ , 这样, 在“*John works while Jane sleeps*”(当约翰工作的时候, 珍妮在睡觉)这个句子中, 从属连接词 *while* 的句法类型便为  $i \setminus i/S$

了。因为在 works 反向展开之后,这个句子的演算过程为:



这样,才可以判定它是一个成立的句子。

这里,((n \ S/i) i) (i \ i/S) n \ S/S, 是因为

i (i \ i/S) i/S

(n \ S/i) (i/S) n \ S/S

由此可见,采用把句法类型反向展开的办法,可以从新的角度对语言现象作出解释。

范畴语法提出已经多年,受到逻辑学界、语言学界的广泛重视,在计算语言学中独具一格,影响深远。在篇章表述理论(Discourse Representation Theory, 简称 RDT),情境语义学(Situation Semantics, 简称 SS),动态谓词逻辑(Dynamic Predicate Logic, 简称 DPL),动态约束理论(Dynamic Binding Theory, 简称 DBT)等计算语言学的语义理论中,都涉及到范畴语法。可以说,范畴语法是学习和研究计算语言学语义理论的必不可少的基本知识,它是当代计算语言学的理论基石之一。

[参考文献]

[ 1 ] J. Lambek, The mathematics of sentence structure, American Mathematical Monthly, Vol. 65, p154—170, 1958.  
 [ 2 ] Y. Bar-Hillel, Decision procedures for structure in natural language, Logique et Analyse, 2e annee, No 5, 1959.  
 [ 3 ] R. Montague, Universal Grammar, in Formal Philosophy, R. H. Thomason (ed.), Yale, New Haven, p222—246, 1970.  
 [ 4 ] Wood, Mary McGee, Categorical Grammar, London, Routledge, 1993.  
 [ 5 ] 冯志伟. 数理语言学简介[J]. 计算机应用与应用数学, 1975. 4。  
 [ 6 ] 冯志伟. 数理语言学[M]. 上海知识出版社, 1985。