载《中国科技术语》，2009 年，第 5 期，Vol.11,No.5

词组型术语结构的自动句法剖析

冯志伟

（教育部语言文字应用研究所）

摘要：本文根据计算术语学的原理，使用短语结构语法对词组型术语进行了自动句法分析。通过具体的实例，介绍了两种基于短语结构语法的剖析方法--自底向上剖析和自顶向下剖析。

关键词：计算术语学，短语结构语法，自动句法剖析，自底向上剖析，自顶向下剖析

中图分类号：N04；H085.2， 文献标识码：A，文章编号：1673-8578(2009)05-0010-07

Automatic syntactic parsing of phrase-term structure

FENG Zhiwei

Abstract: Based on fundamental principle of computational terminology, the phrase-terms are syntactically parsed by the Phrase Structure Grammar (PSG). In this paper, through the concrete examples, the author introduces two kind of parsing approaches: bottom-up parsing and top-down parsing.

Key words: computational terminology, Phrase Structure Grammar (PSG), automatic syntactic parsing, bottom-up parsing, top-down parsing.

我们在《单词型术语的自动分析》[1]一文中，讨论了单词型术语的自动分析问题。在一个术语系统中，词组型术语的数量比单词型术语的数量多得多。因此，在计算术语学（computational terminology）中，除了研究单词型术语的自动分析之外，还有必要研究词组型术语的自动分析问题。词组型术语是由若干个句法单位构成的，是有结构的。词组型术语的自动分析就是计算机自动地识别词组型术语的各个句法单位以及它们之间的相互关系的过程，这个过程，又叫做“自动句法剖析”（automatic syntactic parsing）[2]。

词组型术语的剖析技术是建立在自然语言的形式语法的基础之上的。所谓“剖析”（parsing），就是要用形式语法（formal grammar）来分析词组型术语的结构，使之能清晰地、形式化地表示出来，因此，形式语法在词组型术语的剖析中有着极为重要的作用。

一般地说，一种好的形式语法，在语言的描述方面应该尽量地自然、明白、易懂，在数学的表达方面，应该有很强的说明力和解释力，在计算技术方面，应该具有较高的效率。

美国语言学家乔姆斯基（N.Chomsky）提出，形式语法 G 可以用下面的四元组来定义：

---

[1] 冯志伟，单词型术语结构的自动分析，《中国科技术语》，2009 年，第 2 期。

[2] 冯志伟，一种无回溯的自然语言分析算法，《语言文字应用》，2002 年。

G = (Vn, Vt, S, P)

其中，Vn 是非终极符号的集合，这些符号是专门用来描述语法类别的，它们是范畴符号，如词类符号、词组类型符号等；Vt 是终极符号的集合，它们就是被定义语言中的具体的单词；S 是初始符号，它是集合 Vn 中的一个特殊成员；P 是重写规则的集合，其中的每一条规则都具有

$$\phi \rightarrow \psi$$

的形式，ϕ 称为规则的左部（Left Hand Side，简称 LHS），ψ 称为规则的右部（Right Hand Side，简称 RHS），$\phi \rightarrow \psi$ 意味着可以用规则的右部 ϕ 来置换规则的左部 ψ。

给定了一个语法 G，我们就可以从初始符号 S 开始，应用重写规则推导出这种语法 G 所描述的语言 L(G). 具体地说，我们可以用重写规则 $S \rightarrow \phi_1$，从 S 推导出新的符号串 ϕ_1，再利用重写规则 $\phi_1 \rightarrow \phi_2$，从 ϕ_1 推导出新的符号串 ϕ_2，…，一直到我们得到不能再重写的符号串 ϕ_n 为止。这样推导出的终极符号串 ϕ_n, 就是语言 L(G)中成立的词组型术语。

短语结构语法（Phrase Structure Grammar，简称 PSG）是乔姆斯基形式语法中最重要的一个类型。确切地说，这种短语结构语法应该叫做上下文无关的短语结构语法（Context-Free Phrase Structure Grammar，简称 CF-PSG）。这种语法的重写规则是：

$$A \rightarrow \omega$$

其中，A 是单个的非终极符号（即范畴符号），ω 是非空的符号串，ω 可以由终极符号组成，也可以由非终极符号组成，也可以由终极符号与非终极符号混合组成。

有了一个上下文无关的短语结构语法，我们就可以用 RHS 中的符号串来重写 LHS 中的范畴符号，RHS 的符号串中可以含有范畴符号，也可以含有具体的单词。当用上下文无关的短语结构语法把 LHS 中的范畴符号重写为具体的 RHS 的时候，不必考虑 LHS 的范畴符号所出现的上下文，规则的使用对于上下文没有任何的限制，这就是为什么这种语法叫做“上下文无关的短语结构语法”的原因。当今在程序设计语言中所使用的巴库斯-瑙尔范式（Bacus-Naur Normal Form）就是上下文无关的短语结构语法。

为了行文上的方便，在不引起混淆的情况下，我们在下面的叙述中，把上下文无关的短语结构语法叫做“短语结构语法”。

我们提出如下的短语结构语法来剖析词组型术语“延迟线存储器”和“失灵区部件”：

G = (VN, VT, S, P)
VN = {NP, VP, N, V}
VT = {线，区，部件，存储器，延迟，失灵}
S = {NP}
P:

NP → N	(i)
NP → V + N	(ii)
NP → NP + N	(iii)
VP → V + N	(iv)
N → {线，区，部件，存储器}	(v)
V → {延迟，失灵}	(vi)

词组型术语“延迟线存储器” 和“失灵区部件”的结构是相同的，下面，我们以“延迟线存储器”为例来说明它们的自动过程。

我们从初始状态开始，写出词组型术语“延迟线存储器”的推导过程：

推导过程	所用规则
NP	开始
NP N	(iii)
V N N	(ii)
延迟 N N	(v)
延迟 线 N	(iv)
延迟 线 存储器	(iv)

上述推导过程，也就是这个词组型术语的生成过程。

由短语结构语法生成的词组型术语，可以用如下的树形图来表示：

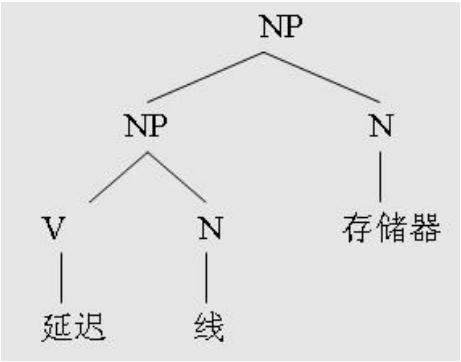

图 1　树形图

这种与短语结构语法相对应的树形图，叫做“短语结构树”（Phrase Structure Tree）。

我们也可以把短语结构树表示为一个表（list），表中的第一个元素是树形图的根上的标记，后面的各个元素是相应结点的直接后裔的标记，按它们在词组型术语中出现的顺序排列，在 LISP 语言中，上述的短语结构树可表示为：

(NP(NP((V 延迟)(N 线))(N 存储器)))

由于表中的第一个元素是树形图中根结点的标记，尔后的各个元素依次是其后裔的标记，而这些元素本身也是表。这样的表写成下面的形式更醒目：

```
(NP
   (NP
      ((V 延迟)
       (N   线)）
           (N   存储器)))
```

如果一种语言可以由短语结构语法来描述，由于短语结构语法是上下文无关的，因此，

这种语言可以称之为上下文无关语言（Context Free Language，简称 CFL）。

由于短语结构语法便于书写，便于修改，因而受到了自然语言处理研究者的普遍欢迎，推动了自然语言处理的发展，在自然语言处理中屡建奇功。短语结构语法的形式清晰，易学易记，在剖析、翻译和编译等技术中得到广泛的应用，自然语言处理早已研制出了用于剖析和识别上下文无关语言 CFL 的高效算法，可见自然语言处理学界对于短语结构语法之重视。词组型术语是自然语言的重要组成部分，我们当然也就可以使用短语结构语法来对词组型术语进行自动的剖析，从而揭示词组型术语的内部结构，加深我们对于词组型术语的认识。因此，在词组型术语的自动句法剖析中，我们也应当重视短语结构语法。

下面，我们以汉语词组型术语为实例，介绍两种基于短语结构语法的剖析方法--自底向上剖析（Bottom-up Parsing）和自顶向下剖析（Top-down Parsing）。

（1）自底向上剖析

如果我们有包含三个词的汉语词组型术语“延迟线存储器”，经过计算机自动切词（automatic segmentation）之后，这个词组型术语的词与词之间出现了空白，其形式变为：

延迟　线　存储器

使用前述的短语结构语法 G，我们可知第一个词“延迟”应该属于 V 这个句法范畴，因为在语法 G 的重写规则(v)中，与规则右部 RHS“延迟”相匹配的规则左部 LHS 是范畴符号 V。这样，我们得到如下的剖析图

V__
延迟　线　存储器

然后，我们继续剖析符号串“ V 线　存储器”。我们检查在语法 G 中，有没有右部 RHS 为 V 的重写规则。例如，如果在语法 G 中有 K→V 这样的重写规则，那么，我们就可以把 V 置于 K 之下，让 K 来支配 V；但是，在我们的语法 G 中没有这样的重写规则，因此，我们来检查所得符号串中的第二个词“线”，根据规则(v)，我们发现“线”的范畴符号是 N，于是，我们得到剖析图

V__　N__
延迟　线　存储器

在剖析过程中，我们要设法在语法 G 所容许的范围内，尽量把符号串中的范畴符号组合起来。首先，我们再一次检查在语法 G 中，有没有右部 RHS 只包含 N 的重写规则，我们发现重写规则(i)正是这样的规则，于是，我们把 N 置于 NP 的支配之下，得到剖析图

NP__
V__　N__
延迟　线　存储器

现在，NP 位于初始符号 V 之后。我们再一次检查语法 G 中有没有右部 RHS 中只包含 NP 的重写规则，检查结果是没有。我们再来检查语法 G 中有没有规则右部 RHS 为符号串 V NP 的重写规则，检查结果也没有，在这种情况下，我们不可能再继续处理了，我们一定是在剖析过程的什么地方误入歧途，而导致了剖析的失败，使剖析进入了死胡同。

为了跳出这个死胡同，我们采用“回溯”（backtracking）的办法，回到剖析过程中进行其他选择的情况中去。为此，我们首先把支配 N 的 NP 去掉，得到剖析图

V__　　N__
延迟　线　　存储器

我们可以看出，前面的剖析过程之所以进入死胡同，是因为我们过早地把 N 置于 NP 的支配之下，而 V　NP 本身又不能单独地出现在语法 G 的重写规则的右部 RHS 之中，因此，我们使用重写规则(iv)，把 V 和 N 置于 VP 的支配之下，我们得到剖析图

____VP_____
V__　　N__
延迟　线　　存储器

在这种情况下，我们首先检查 VP 能否作为规则右部 RHS，发现不行。因此，我们不得不进一步回溯，抹去 VP 这一个范畴符号，于是，我们又得到剖析图

V__　　N__
延迟　线　　存储器

这时我们不再使用重写规则(iv)，而使用重写规则(ii)，把 V 和 N 置于 NP 的支配之下，得到如下的剖析图

____NP_____
V__　　N__
延迟　线　　存储器

我们来检查符号串 NP 能否成为语法 G 中重写规则的右部 RHS，发现不行。于是我们根据规则(v)，把“存储器”规约为 N，得到如下的剖析图：

____NP_____
V__　　N__　　N__
延迟　线　　存储器

再来检查符号串 NP　N 能否成为语法 G 中重写规则的右部 RHS，发现重写规则(iii)正好满足这样的条件，于是，我们把符号串 NP　N 置于 NP 的支配之下，得到如下的剖析图

_____NP_____
____NP_____
V__　　N__　　N__
延迟　线　　存储器

这个 NP 的跨度从词组型术语的头部开始，到词组型术语的尾部结束，覆盖了整个的词组型术语，因此，这个词组型术语的剖析成功。真是：“踏破铁鞋无觅处，得来全不费工夫”。

前面的剖析过程可以归结为如下的搜索树（search tree）：

图 2　自底向上剖析的搜索树

从搜索树上可以看出，要完成一个词组型术语的剖析，其搜索过程是比较复杂的。如果搜索一开始，就能找到正确的途径而得到成功，那当然是最理想不过的。然而，在实际的剖析过程中，往往要经过多次的反复和回溯才能取得成功，有时还要不厌其烦地穷尽各种可能性，我们的程序总有那么一股顽强劲，不达目的，决不休止。在这个搜索树中可以看出，如果我们按照如下的顺序搜索，便可避开死胡同，直接走上成功之途：

延迟　线　存储器
V　线　存储器
V　N　存储器
NP　存储器
N
NP

用 LISP 语言，我们很容易就可以把上述的自底向上剖析过程一目了然地写出来：

(延迟　线　存储器)
((V 延迟)　线　存储器)
((V 延迟)　(N 线)　存储器)
(NP((V 延迟) (N 线))　存储器)
(NP((V 延迟) (N 线))(N 存储器))
(NP(NP((V 延迟)(N 线))(N 存储器)))

心理学家金补尔（J. P. Kimball）研究证明，人们在理解自然语言时，总是试图把新出现是词依附到前面与它紧连的组成成分上，把这个词与它前面的一个词联系起来，以便减轻记忆的负担，避免从记忆中搜索有关的组成成分或词汇。 由于使用这样的策略，人们在理解如下的英语句子时往往会感到困惑：

(1) The man offered one thousand dollars for the conference is my uncle.
（为会议提供一千美圆资助的人是我的叔父）

(2) The horse raced past the barn fell.
（疾驰过牲口棚的那匹马跌倒了）

人们在开始时往往会把句子(1)中的 offered 当作它前面的词 man 的谓语，把句子(2)中的 raced 当作它前面的词 horse 的谓语，等到句子快结束时，才突然发现这样的理解是错误的，于是只好回过头去对句子重新进行分析，采取类似于“回溯”的方法，最后才能得到正确的理解。这种句子叫做“花园幽径句”（garden path sentence），它的理解过程非常曲折，正如花园中曲曲弯弯的路径那样，要屡次三番地重复通过。金补尔这种研究为剖析技术中的回溯机制提供了心理学上的根据。我国学者冯志伟深入研究过英语和汉语中的“花园幽径句”，并且使用计算机对于“花园幽径句”进行了成功的剖析。[3]

（2）自顶向下剖析（Top-down Parsing）

我们仍然以““延迟线存储器”这个词组型术语为例来介绍自顶向下剖析。为了便于读者了解思路，我们以第一人称“我”作为叙述的主体，自顶向下剖析的过程大致如下：

-- 我来找查 NP。
*** -- NP 由什么组成？
-- NP 可以由一个单独的 N 组成。
-- 所以我得首先找查 N。
-- N 由什么组成？
-- 语法 G 中没有什么规则可以扩展 N。
-- 单词“线”可以作为范畴符号 N 的一个成员。
-- “线，区，部件，存储器”这几个单词都是这个词组型术语中开头的第一个词吗？
-- 不是，回溯到***。
-- NP 还可以由什么组成？
-- NP 还可以由一个 V 和一个 N 组成。
-- 现在我需要找查 V。
-- V 由什么组成？
-- V 可以由“延迟，失灵”等单词组成，其中的“延迟”与 V 匹配。
-- “延迟”是这个词组型术语中开头的第一个词吗？
-- 是的。
-- 现在我发现 V 是由单词“延迟”组成的，我需要继续找查 V 后面的 N。
-- N 由什么组成？
-- N 可以由“线，区，部件，存储器”等单词组成，其中的“线”与 N 匹配。
-- “线”是这个词组型术语中开头的第二个词吗？
-- 是的。
-- 现在我发现这个 N 是由单词“线”组成的，因此， NP 是由包含单词“延迟”的一个 V 以及包含单词“线”的一个 N 组成的。

[3] 冯志伟，花园幽径句的自动分析算法，《当代语言学》，2003 年，第 4 期。

-- 是不是到达词组型术语“延迟线存储器”的结尾了？

-- 没有，剖析还没有结束。

-- 哎呀，一定是我做错了什么事。

-- 回溯到***处，用另外的办法来做。

-- NP 还可以由什么组成？

-- NP 还可以由另一个 NP 和一个 N 组成。

-- 另一个 NP 可以由什么组成？

-- 另一个 NP 还可以由一个 V 和一个 N 组成。

-- 现在我需要找查 V。

-- V 由什么组成？

-- V 可以由“延迟，失灵”等单词组成，其中的“延迟”与 V 匹配。

-- “延迟”是这个词组型术语中开头的第一个词吗？

-- 是的。

-- 现在我发现 V 是由单词“延迟”组成的，我需要继续找查 V 后面的 N。

-- N 由什么组成？

-- N 可以由“线，区，部件，存储器”等单词组成，其中的“线”与 N 匹配。

-- “线”是这个词组型术语中开头的第二个词吗？

-- 是的。

-- 现在我发现这个 N 是由单词“线”组成的，因此， NP 是由包含单词“延迟”的一个 V 以及包含单词“线”的一个 N 组成的。

-- 是不是到达词组型术语的结尾了？

-- 没有，它后面还有一个 N。

-- N 由什么组成？

-- N 可以由“线，区，部件，存储器”等单词组成，其中的“存储器”与 N 匹配。

-- 我发现这个 N 是由单词“存储器”组成的，所以，整个的词组型术语“延迟线存储器”是由 NP 和它后面的一个 N 组成的。

-- 是不是到达词组型术语的结尾了？

-- 是的，已经达到词组型术语的结尾。

-- 剖析成功了！

图 3 是“延迟线存储器”自顶向下剖析的搜索树。在自顶向下的识别过程中，某一时刻的情况可用两个序列来描述：一个序列由剖析目标组成，一个序列由剩下的单词组成，两个序列之间用冒号(:)隔开。例如，“延迟线 N：存储器”说明，自顶向下剖析程序试图找出跟着“延迟线”的 N，而这时剩下的单词是“存储器”。

图 3　自顶向下剖析的搜索树

在这搜索树中，如果我们按照如下的顺序搜索，便可得到成功：

剖析目标	剩下的单词序列
NP :	延迟 线 存储器
NP N :	延迟 线 存储器
V N N :	延迟 线 存储器
延迟 N N :	延迟 线 存储器
N N :	线 存储器
线 N :	线 存储器
N :	存储器
存储器 :	存储器
:	

（剖析成功!）

自顶向下剖析程序的写法与自底向上剖析程序的写法十分相似。自底向上剖析程序须要了解在任何特定的时刻成功地找查到的东西是什么，而自顶向下剖析程序则须要记住它试图要找查的东西是什么，这就是它的剖析目标（goals）。因此，自顶向下剖析程序在任何一个时刻的情况可用目标序列及单词序列来描述。在 LISP 语言中，目标序列用表（list）来表示，单词序列用原子（atom）来表示。当我们调用自顶向下剖析程序时，目标序列的表就是((NP))，它指出这个剖析程序所要找查的表就是这个(NP)，而单词序列的原子所组成的表（list of atoms）就是输入的词组型术语，这个词组型术语中的单词要满足目标的要求。当目标序列与单词序列两者都变空时，自顶向下剖析就获得成功，输入句子就得到识别。自顶向下剖析过程可用 LISP 语言表示如下：

目标序列	单词序列
(NP)	(延迟 线 存储器)
((NP)(N))	(延迟 线 存储器)
(((V)(N))(N))	(延迟 线 存储器)
(((延迟)(N))(N))	(延迟 线 存储器)
(((N))(N))	(线 存储器)
(((线))(N))	(线 存储器)
((N))	(存储器)
(存储器)	(存储器)
( )	( )

在剖析过程中，如果目标序列的表中的第一项是范畴符号，那么，我们就用语法中与这个范畴符号匹配的规则的右部来替换这个范畴符号；如果目标序列的表中的第一项是单词，而这个单词与单词序列中的第一个单词相同，那么，就把这两个单词都抹掉，并继续进行找查，直到目标序列的表与单词序列两者都变空为止。

词组型术语“失灵区部件”的结构是

(NP(NP((V 失灵)(N 区))(N 部件)))

其结构与“延迟线存储器”相同，其剖析过程也是一样的。读者可以自己把“失灵区部件”这个词组型术语的剖析过程写出来。因为我们的剖析方法具有普遍性，这个方法对于各种类型的词组型术语的自动剖析都是适用的。

在上面使用过的短语结构语法的规则中，“NP → NP + N”这个规则的左部是 NP，右部也出现了同样的 NP，这样，在剖析的过程中，左部的 NP 会嵌入到右部的 NP 中去，形成“左递归”（left-recursive）的现象。

NP → NP + N
NP → NP + N + N
NP → NP + N + N + N
…….

这样，程序就会不断地运行下去，永远不会休止。这显然是很危险的。

为了避免出现左递归，我们可以把 规则右部的 NP 改成 Nominal（名词性成分），使它与规则左部的 NP 区别开来。这是程序技术中比较复杂的问题，我们就不再讨论了。

此外还有一些效率比较高的剖析技术，限于篇幅，兹不赘述。

由以上的介绍可以看出，计算术语学是一门很有价值的新学科，它需要我们把术语学、语言学和计算机科学巧妙地结合起来。希望我国术语学界关注这个新学科，推动我国术语学的发展。

参考文献

1. 冯志伟，自然语言的计算机处理，上海外语教育出版社，1996 年。
2．冯志伟，形式语言理论，《计算机科学》，1979 年，第 1 期，p34-57。
3．冯志伟，数理语言学，知识出版社，1985 年。
4. 冯志伟，计算术语学，《术语标准准化与信息技术》，2008 年第四期，总第 52 期，p4-9.
5. D. Jurafsky 等著，冯志伟等译，自然语言处理综论，电子工业出版社，2005 年。
6. Christian Jacquemin, Spotting and Discovering Terms through Natural Language Processing, The MIT Press, 2001.