载《语言科学》，第 5 卷，第 3 期，p14-23，2006 年 5 月

用上下文无关语法来描述汉字结构[1]

冯志伟

教育部语言文字应用研究所　　北京　　100010

提要　上下文无关语法(简称 CFG)在自然语言的句法自动分析中已经得到广泛的应用。本文使用这种 CFG 语法来分析和描写汉字的结构，以部件作为汉字结构的枢纽，把汉字部件的 11 种结构方式看成 CFG 中的非终极符号，把末级部件看成 CFG 中的终极符号，使用树形图及其等价的括号表达式成功地对汉字的结构进行了形式描述。

关键词　上下文无关语法　部件　末级部件　汉字结构　树形图　括号表达式

1　部件是汉字结构的枢纽

汉字可以分为独体字（single character）和合体字（compound character）两类。独体字在字形结构上分解不出几个相离的部件而只能分解出笔画。例如，“甘、手、亦”。合体字是由两个或两个以上的部件组合而成的字。例如“休”字由“亻”和“木”两个部件组成，“霜”由“雨”、“木”和“目”三个部件组成。合体字的结构可以分为三个层次：第一个层次是合体汉字本身，第二个层次是组成这个合体字的部件（component），第三个层次是组成部件的笔画（stroke）。

在汉字形体结构的三个层次中，部件是枢纽性的一环，是汉字形体结构的核心。

从汉字的发展历史来看，古人有所谓“独体为文，合体为字”的说法。东汉许慎在《说文解字》第十五中说：“仓颉之初作书，盖依类象形，故谓之文，其后形声相益，即谓之字。文者，象物之本，字者，言孳乳而浸多也。”[1]在这段话中，许慎明确地把“文”与“字”区别开来，认为“字”是由象形的“文”孳乳繁衍出来的。可见，造字的历史，不是先造笔画，再造部件，然后再造出整个的汉字，而是先造出了一些象形的“文”，这些“文”又作为部件繁衍而成为数量众多的合体的“字”。例如，由“日”和“月”这两个象形的文，用会意的方法合成“明”这个合体字。至于笔画系统的形成，那是进入隶书阶段以后的事情，汉字形体的新陈代谢是一种笔势的变革，除了草书和简化字之外，基本上不是结构本身的变革。由此可见，由单体的“文”到合体“字”，是合乎汉字发展规律的，所以，研究汉字的形体结构，应当从部件入手，这样才可能抓住问题的关键，做到纲举目张。

从汉字的现状来看，自从汉字形成了平直的笔画系统和方正的方块体系之后，部件具有承上启下的作用，它一方面由几个简单的笔画构成，另一方面又进一步构成成千上万个汉字。部件处于枢纽的地位，把笔画与汉字联系起来，成了汉字形体结构的核心。部件总数只有几百个，合成一个汉字时，只需要为数不多的几个部件，把部件分解为笔画，得出的笔画数目也不多。如果我们不要部件这个承上启下的枢纽，直接把汉字分解为笔画，所得的笔画数目会很多，排列起来犹如一个长蛇阵，不容易说清楚其中的序列关系。例外，部件具有固定的形体，大多数部件有一定的含义，许多部

[1] 本文得到国家社会科学基金（项目批准号 03BYY019）的资助。

件还有明确的称读，在性质上与汉字比较接近，有的部件同时也就是独体汉字，远较笔画优越。

因此，部件应该是汉字形体结构研究的中心内容。我们在研究汉字结构的形式化描述时，一定要充分重视部件的这种枢纽作用。

2 汉字结构与上下文无关语法

如果一个部件再继续分解就成为笔画，那么，就说明这个部件已经不能再继续分解为更小的部件，这样的部件叫做末级部件（primitive component）。独体字再继续分解就是笔画，因此，独体字也可以看成是由一个末级部件组成的汉字。从这个意义上说，所有的汉字（包括独体字与合体字）都是由末级部件组合而成的。

例如，“霜”字首先可以分解为“雨”和“相”两部分，其中，“雨”是独体字，属于末级部件，再继续分解就成了笔画，而“相”是合体字，还可以进一步分解为末级部件“木”和“目”。我们可以用树形图（tree graph）表示如下：

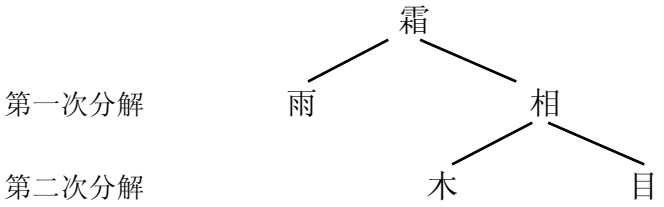

图 1. “霜”字的结构的树形图表示

我们注意到，在图 1 中，“雨，木，目”等末级部件都出现在树形图的叶子结点上，而“霜，相”则不出现在树形图的叶子结点上。

对于每一个合体汉字，我们都可以使用这样的办法，把它分解为树形图，从而揭示出它的结构。这对于汉字信息处理和对外汉语教学，显然是很有好处的。

根据计算机统计的结果，现代汉字是由 648 个末级部件组成的。在这 648 个末级部件中，有 327 个是独体字，如“口，木，土，十，又”等，另外的 321 个不是独体字，它们仅仅是合体字的组成单位，如“彡，犭，扌，辶，忄，亻，钅”等。如果我们掌握了这 648 个末级部件，再掌握了合体字的分解方法，那么，我们在汉字信息处理和对外汉语教学中，就可以达到以简驭繁事半功倍的效果。[2].

Chomsky 的上下文无关语法（Context Free Grammar，简称 CFG）是自然语言处理中应用得最为广泛的一种形式语法[3]，这个语法在数学上简洁、清晰，在语言学上比较好的解释力，在程序的实现上有比较成熟的算法，受到计算语言学研究者的欢迎。

我们是不是也可以采用上下文无关语法来描述汉字的结构呢？答案是肯定的。对于国家标准 GB2313-80 中的 6763 个汉字的结构进行分析之后，我们发现，汉字的结构完全可以使用上下文无关语法来描述。

根据 Chomsky 的形式语言理论，一个上下文无关语法 G 可以用四元组来表示。这个四元组可以定义如下：

G = (Vn, Vt, S, P)

其中。Vn 是非终极符号的集合，它们不出现在语法树形图的叶子结点上，Vt 是终极符号的集合，它们只出现在语法树形图的叶子结点上，S 是初始符号，它出现在语法树形图的根结点上，初始符号也是一种非终极结点，P 是重写规则，其形式为：

$$A \rightarrow \omega$$

这里，A 是单独的非终极符号，ω 是符号串，它可以由终极符号或非终极符号组成。

在图 1 中，出现在树形图的叶子结点上的“雨，木，目”相当于上下文无关语法中的终极结点，不出现在树形图叶子结点上的“霜，相”相当于上下文无关语法中的非终极结点，“霜”出现在树形图的根结点上，而根也是非终极结点。由于“霜，相”不出现在树形图的叶子结点上，它们的结构表示了某种信息，“霜”是由“雨”和“相”上下相接而组成的，它表示了“上下结构”这样的结构方式信息（construction pattern），“相”是由“木”和“目”左右相接而组成的，它表示了“左右结构”这样的结构方式信息。

因此，我们可以写出如下的重写规则：

霜（上下结构） → 雨 + 相（左右结构）

相（左右结构） → 木 + 目

由于“霜”和“相”都不出现在树形图的叶子结点上，所以，在重写规则中，我们只需要写出它们所代表的结构方式信息，这样，上述的重写规则可以改写为：

上下结构 → 雨 + 左右结构

左右结构 → 木 + 目

这两个规则的左部分别是“上下结构”和“左右结构”，它们都是单独的非终极符号，与上下文无关语法的重写规则 A → ω中的左部 A 相对应，第一个规则的右部是“雨 + 左右结构”，“雨”是末级部件，属于终极符号，“相”是非终极符号，它们是由终极符号和非终极符号组成的符号串，与上下文无关规则 A → ω中的右部ω相对应。

这样一来，我们就可以用上下文无关语法来描述“霜”字的结构了。这个上下文无关语法可以这样来写：

G = (Vn, Vt, S, P)

其中，

Vn = {上下结构，左右结构}

Vt = {雨，木，目}

S = {上下结构}

P:

上下结构 → 雨 + 左右结构

左右结构 → 木 + 目

“上下结构”“左右结构”都是表示范畴的概念，我们可以使用符号来代表它们。例如，我们可以使用符号 A 代表“上下结构”，用符号 C 代表“左右结构”，那么，上面的上下文无关语法可以写得更加简洁：

G = (Vn, Vt, S, P)

其中，

Vn = {A，C}

Vt = {雨，木，目}

S = {A}

P:

A → 雨 + C

C → 木 + 目

显而易见，从上下文无关语法的角度来看汉字，Vn 就是汉字的结构方式，Vt 就是构成汉字的末级部件，S 就是需要分解的汉字的最顶一级的结构方式，P 就是分解的规则，由于 P 的左部是一个单独的非终极符号，右部是一个符号串，因此，这样的

语法完全符合 Chomsky 关于上下文无关语法的定义。

汉字的结构方式 Vn 究竟有多少？经过统计分析证实，汉字的 Vn 是有限的，一共有如下 11 种：

⑴上下结构，记为 A。

例如： 志，呆，苗，字。

⑵上中下结构，记为 B。

例如，曼，禀，复，享。

⑶ 左右结构，记为 C。

例如，伟，亿，课，化。

⑷ 左中右结构，记为 D。

例如，衍，棚，树，狱。

⑸ 左上包围结构，记为 E。

例如，庙，病，房，尾。

⑹ 右上包围结构，记为 F。

例如，句，氧，可，习。.

⑺ 左下包围结构，记为 G。

例如，达，旭，连，爬。.

⑻ 左上右包围结构，记为 H。

例如，同，问，闹，风。

⑼ 上左下包围结构，记为 I。

例如，区，医，匿，匣。

⑽ 左下右包围结构，记为 J。

例如，凶，画，击，函。

⑾ 全包围结构，记为 K。

例如，困，国，回，团。

此外，还有一种特殊的对称结构，例如，米，韭，隶，垂。 这样的结构不能进一步拆分，从结构分析的角度来看，它们的性质与独体字、末级部件是一样的，属于不能再进一步分解的结构，因此，我们把这些不能进一步分解的结构都记为 O，由于它们都不能进一步分解，应该属于终极符号 Vt，在树形图中它们都处于叶子的位置上，用符号■表示。

根据这样的分析，在表示汉字结构的上下文无关语法中，非终极符号 Vn 就是 A，B，C，D，E，F，G，H，I，J，K 等 11 个符号，它们表示了汉字的基本结构方式，终极符号 Vt 就是 648 个末级部件，包含独体字（如“口，木，土，十，又”等），偏旁部首（如“纟，犭，扌，辶，忄，亻，钅”等）以及对称结构字（如“米，韭，隶，垂”等）。

这样一来，我们可以把表示汉字结构的上下文无关语法重新定义如下：

G = (Vn, Vt, S, P)

其中，

Vn = {A, B, C, D, E, F, G, H, I, J, K}

Vt = {O}

O 可以为各种终极符号。例如，口，木，土，十，又，纟，犭，扌，辶，忄，亻，钅，米，韭，隶，垂…等等。

初始符号 S 就是要分析其结构的汉字本身，它可以取 Vn 中的符号为其值。当我

们用上下文无关语法来分析自然语言的句子的时候，初始符号 S 只有一个，这就是表示句子(Sentence)的符号 S，而在用上下文无关语法描述汉字的时候，初始符号 S 可以取 Vn 中的符号为其值，可以有 11 种不同的选择，这意味着，对于汉字描述来说，我们总是有 S∈Vn，S 可以取 Vn 中的不同的值。但是，在用上下文无关语法描述句子的时候， S 只能取 Vn 中惟一的一个值（也就是 Sentence 的缩写表示 S 这个非终极符号本身），只有一种选择。这是用上下文无关语法来描述汉字结构时与描述句法结构是的差别。

重写规则 P 具有上下文无关语法的规则的形式，它的左部必须是一个单独的非终极符号，右部是一个符号串，其形式是：

$$A \rightarrow \omega$$

例如，

A → 雨 + C　　（ⅰ）

C → 木 + 目　　（ⅱ）

使用重写规则(i)和(ⅱ),我们可以写出“霜”字的推导史(derivational history)如下：

A　　（初始符号）

雨 + C　　（使用规则ⅰ）

雨 + 木 + 目　　（使用规则ⅱ）

这样，我们也可以把一个汉字看成是从初始符号开始，使用上下文无关语法的重写规则，一步一步生成的，也就是把汉字看成是由上下文无关语法生成的结果。

在上面的图 1 所表示的树形图中，“霜”是上下结构，“相”是左右结构，我们可以在树形图中的相应结点上加上“上下结构”和“左右结构”的标记，“雨，木，目”都是末级部件，用不着再加其他标记了。这样，图 1 中的树形图可以改写如下：

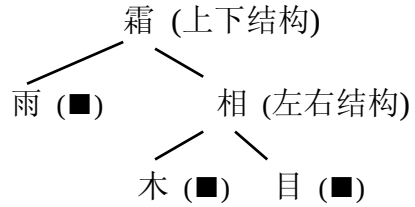

图 2. 加了标记的树形图

这个树形图又可以进一步抽象如下：

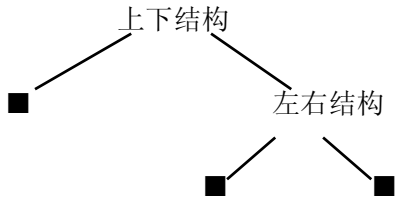

图 3. 树形图的进一步抽象

使用上下文无关语法中约定的终极符号和非终极符号，树形图可以表示为：

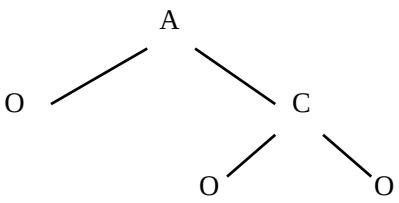

图 4. 结点用终极符号和非终极符号表示的树形图

这个树形图可以转写成等价的括号表达式（bracket formula）：

A(O, C(O, O))

这意味着，合体字“霜”可以表示为括号表达式 A(O,C(O,O))。

任何一个合体字都可以表示为这样的括号表达式。这样，我们便找到汉字结构的一种形式化表示方法—括号式表示法。这是一种基于上下文无关语法的表示方法[4]。

3 合体字结构的形式描述

很多常用汉字不只包含两个或三个部件，往往包含多个部件，因此，我们必须对它们逐级分解。下面，我们说明包含三个或三个以上部件的合体字的结构分解情况，这样的结构分解也就是对于它们的形式描述。

3.1 包含三个部件的合体字的形式描述

包含三个部件的合体字按照形式可以分为 15 个小类。

⑴ A(O,C(O,O))

与这个括号式相应的树形图为：

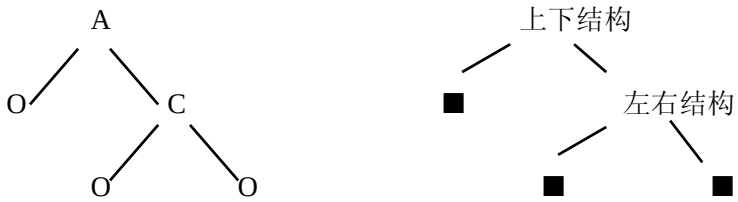

图 5. 与括号式 A(O,C(O,O))相应的树形图

例如，“花”可以表示为如下的树形图：

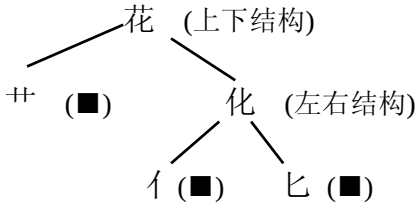

图 6. “花”字的树形图

“花”的结构与“霜”属于同一个小类，都是 A(O,C(O,O))。

此外的小类还有，括号式如下：

⑵ C(O,A(O,O))：例如，“陪”。

⑶ C(O,E(O,O))：例如，“缠”。

⑷ C(O,G(O,O))：例如，“挺”。

⑸ C(O,H(O,O))：例如，“润”。

⑹ C(O,I(O,O))： 例如，“抠”。

⑺ C(O,K(O,O))：例如，“捆”。

⑻ E(O,A(O,O))：例如，“庶”。

⑼ E(O,C(O,O))：例如，“厢”。

⑽ H(O,A(O,O))：例如，“闾”。

⑾ K(O,A(O,O))：例如，“圄”。

⑴ → ⑾ 的结构可以归纳为如下的几何形式：

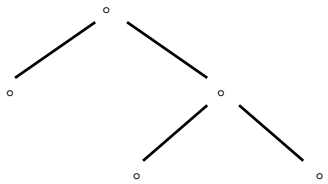

图 7. ⑴ → ⑾具有相同的几何形式

⑿ A(C(O,O),O)

相应的树形图为：

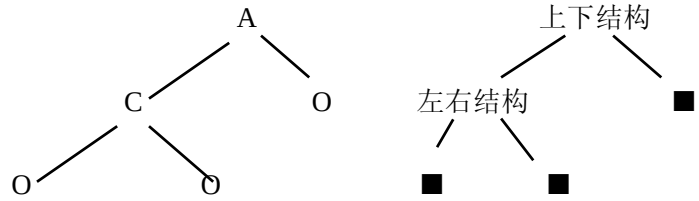

图 8. 与括号式 A(C(O,O),O)相应的树形图

例如，合体字“型”的结构如下：

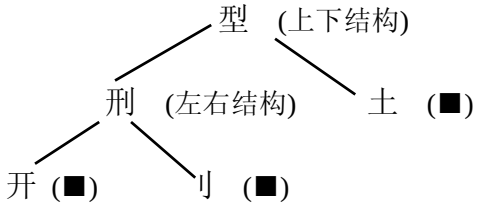

图 9. “型”字的树形图

⒀ C(A(O,O),O)：例如，“部”。

⒁ G(A(O,O),O)：例如，“逞”。

⒂ G(C(O,O),O)：例如，“逊”。

⑿ → ⒂ 的结构可以归纳为如下的几何形式：

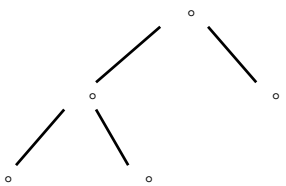

图 10. ⑿ → ⒂具有相同的几何形式

3.2 包含四个部件的合体字的形式描述

可分为 19 个小类：

⑴ C(O,A(O,C(O,O)))

树形图为：

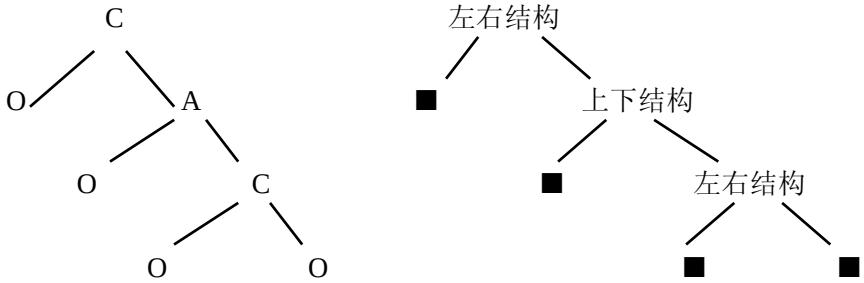

图 11. 与括号式相应的树形图

例如，“摄”可以表示为如下的树形图：

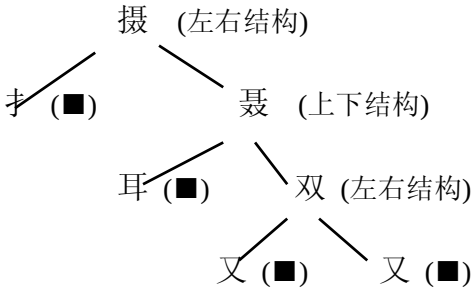

图 12. “摄”字的树形图

此外的小类还有，括号式如下：

⑵ A(O,C(O,A(O,O)))：例如，“寤”。

⑶ A(O,A(O,C(O,O)))：例如，“蕊”。

⑷ C(O,I(O,C(O,O)))：例如，“櫂”。

⑸ H(O,C(O,A(O,O)))：例如，“阔”。

⑹ I(O,A(O,E(O,O)))：例如，“匿”。

⑴→⑹ 的结构可以归纳为如下的几何形式：

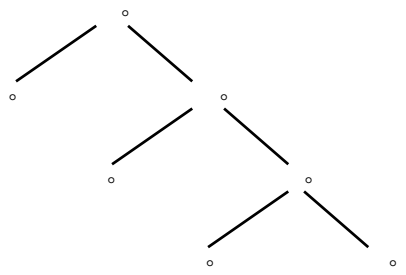

图 13. ⑴→⑹的几何形式

⑺ C(A(O,O),A(O,O))

树形图为：

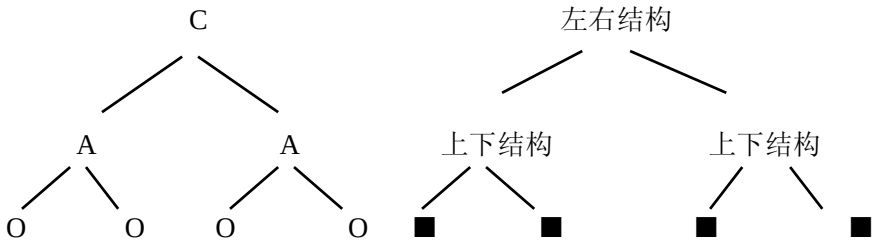

图 14. 与括号式 C(A(O,O),A(O,O))相应的树形图

例如，“韶”字可表示为如下的树形图：

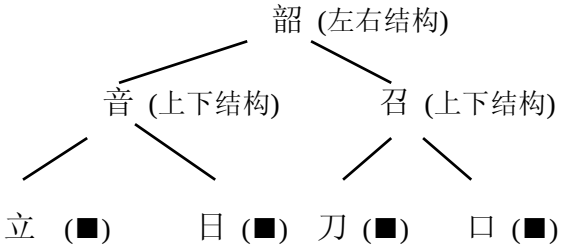

图 15. “韶”字的树形图

此外的小类还有，括号式如下：

⑻ A(C(O,O),I(O,O))：例如，“筐”。

⑼ A(C(O,O),E(O,O))：例如，“蒌”。

⑽ C(I(O,O),A(O,O))：例如，“欧”。

⑺→⑽结构可归结为如下的几何形式：

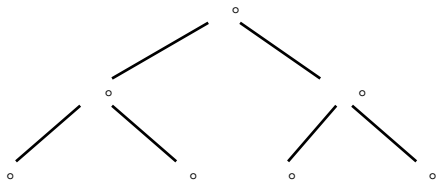

图 16. ⑺→⑽的几何形式

⑾ B(O,O,A(O,O))

树形图为：

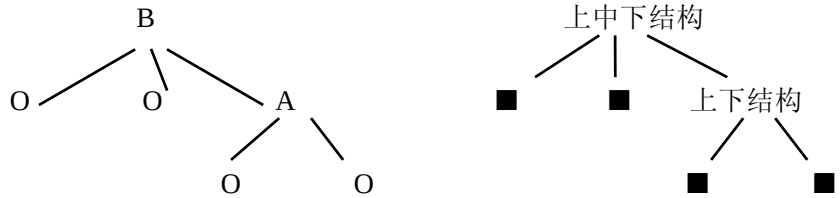

图 17. 与括号式 B(O,O,A(O,O))相应的树形图

例如，“营”字可表示为如下的树形图：

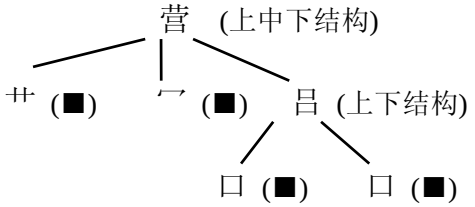

图 18. “营”字的树形图

此外的小类还有，括号式如下：

⑿ D(O,O,A(O,O))：例如，“游”。

⑾→⑿ 结构可以归结为如下的几何形式：

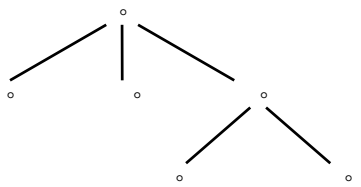

图 19. ⑾→⑿的几何形式

⒀ C(B(O,O,O),O)

树形图为：

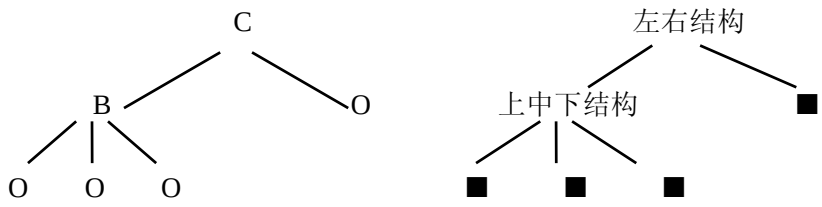

图 20. 与括号式 C(B(O,O,O),O)相应的树形图

例如，“额”字可表示为如下的树形图：

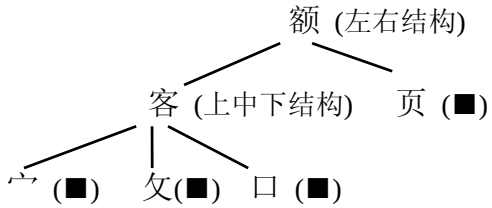

图 21. “额”字的树形图

此外的小类还有，括号式如下：

⑭ A(D(O,O,O),O)：例如，“鬻”。

⑬→⑭结构可以归结为如下形式：

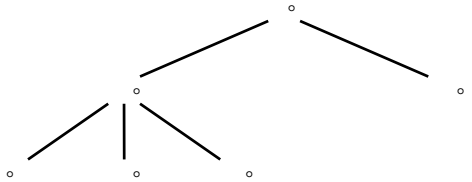

图 22. ⑬→⑭的几何形式

此外的小类还有，括号式如下：

⑮ C(O,B(O,O,O))

树形图为：

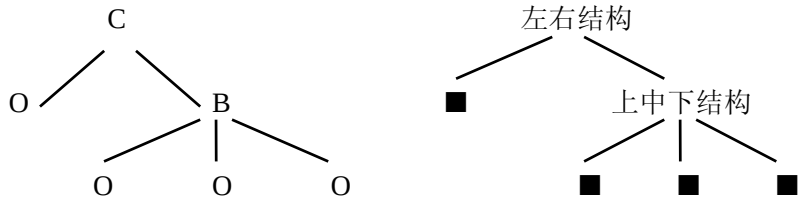

图 23. 与括号式 C(O,B(O,O,O))相应的树形图

例如，“樟”字可表示为如下的树形图：

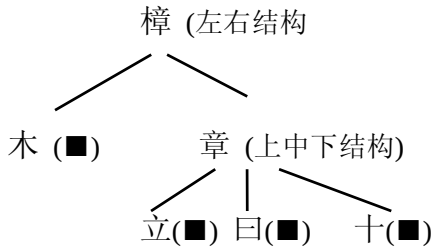

图 24. “樟”字的树形图

其几何形式抽象为如下形式：

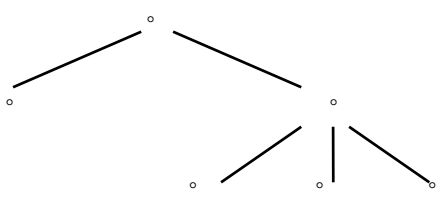

图 25. ⑮的几何形式

此外的小类还有，括号式如下：

⑯ A(O,A(C(O,O),O))

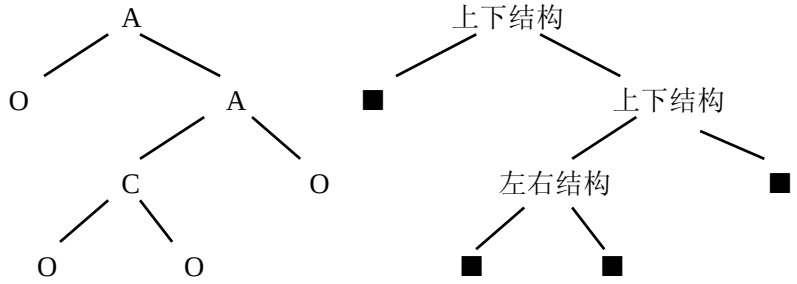

图 26. 与括号式 A(O,A(C(O,O),O))相应的树形图

例如，“莘”字可表示为如下的树形图：

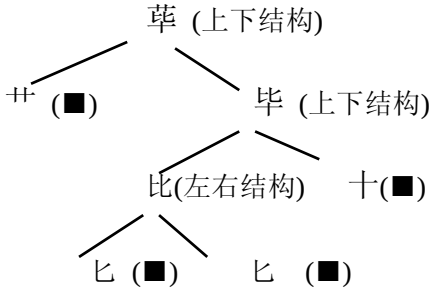

图 27. “莘”字的树形图

此外的小类还有，括号式如下：

⑰ C(O,A(O,O),O))：例如，“燃”。

⑱ E(O,A(C(O,O),O))：例如，“腐”。

⑯→⑱结构可归结为如下形式：

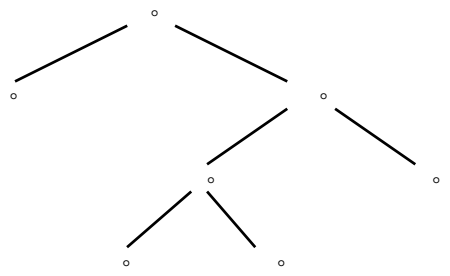

图 28. ⑯→⑱的几何形式

此外，还有：

⑲ G(E(O,A(O,O)),O)：例如，“遮”。

通过对于上面的汉字结构的分析，想来读者已经理解了我们的方法，我们建议读者自己做出这个汉字的树形图。

在下面的论述中，为了节省篇幅，我们不再做树形图，只写出汉字的括号式结构。

3.3 包含五个部件的合体字的形式描述

包含五个部件的合体字可分为 19 个小类，其形式描述用括号式表示如下。

⑴ C(O,B(O,C(O,O),O))：例如，“澡”。

⑵ A(O,B(O,A(O,O),O))：例如，“膏”。

⑶ C(O,B(O,O,H(O,O)))：例如，“搞”。

⑷ A(O,B(O,O,H(O,O)))：例如，“嵩”。
⑸ C(O,A(C(O,O),C(O,O)))：例如，“缀”。
⑹ E(O,A(C(O,O),G(O,O)))：例如，“魔”。
⑺ D(O,B(O,O,O),O)：例如，“渤”。
⑻ C(B(O,O,H(O,O)),O)：例如，“敲”。
⑼ A(D(O,A(O,O),O),O)：例如，“樊”。
⑽ B(C(O,O),O,C(O,O))：例如，“嚣”。
⑾ C(A(O,D(O,O,O)),O)：例如，“酃”。
⑿ C(A(O,O),B(O,O,O))：例如，“蹊”。
⒀ C(A(C(O,O),C(O,O)),O)：例如，“戳”。
⒁ A(C(O,O),A(C(O,O),O))：例如，“篮”。
⒂ A(O,C(O,B(O,O,O)))：例如，“寝”。
⒃ C(O,E(O,A(O,C(O,O)))：例如，“漉”。
⒄ C(B(O,O,O),A(O,O))：例如，“毅”。
⒅ B(O,O,D(O,O,O))：例如，“赢”。
⒆ A(O,G(E(O,A(O,O)),O))：例如，“蘧”。

3.4 包含六个部件的合体字的形式描述

包含六个部件的合体字可分为 10 个小类，其形式描述用括号式表示如下：
⑴ C(A(F(O,O),F(O,O)),A(O,O))：例如，“歌”。
⑵ A(C(I(O,O),A(O,O)),C(O,O))：例如，“翳”。
⑶ C(B(O,O,O),B(O,O,O))：例如，“豁”。
⑷ A(C(O,O),E(O,A(O,C(O,O)))：例如，“麓”。
⑸ C(B(O,O,O),A(O,C(O,O)))：例如，“豌”。
⑹ A(C(E(O,A(O,O)),A(O,O)),O)：例如，“臀”。
⑺ C(O,B(O,O,D(O,O,O)))：例如，“瀛”。
⑻ D(O,A(C(O,O),C(O,O)),O)：例如，“衢”。
⑼ C(O,B(C(O,O),O,A(O,O)))：例如，“骥”。
⑽ C(O,B(O,C(O,O),C(O,O))：例如，“灌”。

3.5 包含七个部件的合体字的形式描述

包含七个部件的合体字可分为 4 个小类，其形式描述用括号式表示如下：
⑴ A(C(B(O,O,O),B(O,O,O)),O)：例如，“戆”。
⑵ C(E(O,A(O,C(O,O))),A(O,C(O,O)))：例如，“麟”。
⑶ A(C(A(O,O),E(O,A(O,O))),A(O,O))：例如，“饕”。
⑷ C(A(O,O),B(O,C(O,O),A(O,O)))：例如，“儴”。

3.6 包含八个和九个部件的合体字的形式描述

包含八个部件的合体字只有 1 个小类，包含九个部件的合体字也只有 1 个小类，其形式描述用括号式分别表示如下：
包含八个部件的合体字的括号式为：
C(B(O,O,O),B(O,C(O,O),A(O,O)))。。例如，“鬱”。
包含九个部件的合体字的括号式为：

C(B(O,O,B(O,O,O)),A(C(O,A(O,O)),O))。例如，“懿”。

4 小结

本文使用上下文无关语法来描述汉字的结构，采用类似于句法分析的方法，把汉字或者表示为一个树形图，或者表示为与之相应的括号式，这样，我们就可能对汉字进行自动分析或处理，从而推动中文信息处理的研究。当然，这样的方法对于面向非汉族人的汉字教学也是有好处的。

我们的目的只在于提出汉字形式描述的方法，并不试图穷尽地列举出全部汉字的形式结构。由于本文对于汉字的分析是在 GB-2312-80《信息交换用汉字编码字符集—基本集》6763 个汉字的范围内进行的，如果进一步扩大汉字字符集的规模，汉字形式结构的类型和数量必定还会增加。

参考文献

[1] 许慎 1963 《说文解字》，中华书局出版。

[2] 冯志伟 1989 《现代汉字和计算机》，北京大学出版社。

[3] N. Chomsky 1956 Three models for the description of language, *IRI Transactions on Information Theory*, 2(3), 113-124.

[4] Feng Zhiwei 1994 *Die chinesiscnen Schriftzeichen in vergangenheit und Gegenwart*（德文）, Germany: Wissenschaftlicher Verlag Trier.

作者简介

冯志伟，男，1939 年 4 月生，云南昆明人。先后在北京大学和中国科技大学研究生院获双硕士学位，教育部语言文字应用研究所学术顾问，中国科学院国家模式识别重点实验室学术委员会委员，博士生导师。主要从事计算语言学研究，出版《数理语言学》、《现代语言学流派》、《机器翻译研究》、《自然语言的计算机处理》等著作 20 部。

Description of Chinese character structure by Context Free Grammar

Feng Zhiwei

Institute of Applied Linguistics (Ministry of Education) Beijing 100010

Abstract: The context free grammar (CFG) was applied broadly in the automatic syntactic parsing of natural language processing. We use CFG to describe and analyze Chinese character structure. In the description and analysis of Chinese character, we regard the component of Chinese character as kernel pivot of Chinese character structure, then we take 11 construction patterns of Chinese character as non-terminal symbols of CFG, and we take the primitive components of Chinese character as terminal symbols of CFG. The structure of Chinese character is represented by the tree graph or its equivalent bracket formula.

Keywords: context free grammar, component, primitive component, Chinese character structure, tree graph, bracket formula