**FCG中的设计方法**

Luc Steels, 2012

1301211280 田骏

**翻译部分**

**一、简介**

FCG是近年来一系列基于认知语言学和构式语法而提出的构想（idea）的形式化实践。一个构式是语言中的一个普通的使用模式（a regular pattern of usage），如单词、词的组合、习语，或具有约定的意义和表达功能的句法模式（syntactic pattern）。

构式语法则把语言中不同的构式编纂成目（catalogs），兼顾其语义和句法两个方面。虽然构式语法通常只停留在口头讨论层面（described only in verbal terms），特别是在第二外语教学的应用中，但我们仍可以将构式语法形式化及操作化，以对自然语言的处理过程建模。这样的实现方式清楚地表达了一个构式的内涵（entails），并使得构式语法可以应用于计算领域。

构式语法的形式化与生成语法主要有两个方面的不同：

1. 构式的定义采用了形式和意义的双向绑定（bi-directional associations）[[1]](#footnote-1)，因此同一个构式可以不加修改地应用于语言理解和语言生产中而不需要任何折中妥协[[2]](#footnote-2)。这里的语言生产指的不仅仅是随机地生成一个可能的句子，而是经概念化（conceptualization）得来的意义在既有的语言约定下生成的最好的句子。

2. 双向绑定需要考虑语言的方方面面（语用、句法、语义、形态、语音），因为人类语言并不是模块化的（modularly organized）[[3]](#footnote-3)。（举了一个Hungarian复数谓词一致性与五个层次均相关的例子。）因此，词汇和语法[[4]](#footnote-4)是综合语言不同层次纵向组织的，而非依靠独立于语义和语音之外的句法模块横向地自治地组织。

因此，在形式化的构式语法中的构式不仅定义了句法模式，还有该模式蕴含的语义结构，并可能包含语用、形态和语音方面，或是额外的构式义[[5]](#footnote-5)。

FCG是逐渐增长的计算构式语法族中的一员，该族还包括ECG，SBCG等。其目的是建立一个深层次的语言生产和语言理解系统，作为语义接地的（grounded）人-机或机-机交互的核心，以利用由知觉和感觉派生[[6]](#footnote-6)的世界模型[[7]](#footnote-7)。深层语言处理则要求能处理大量的语法结构表征并将语义很好地整合进去[[8]](#footnote-8)（这与将语义分析的任务委派给另一个组件相对）。

FCG吸收了许多计算语言学家熟悉的技术（特征结构、合一运算等）。同时，FCG引入了一系列创新，例如J-算子。FCG基于LISP语言实现，并已应用于一系列有关语言进化或者人机交互的实验中。

现在已有许多文献介绍FCG成功的实例。本文着重介绍FCG的设计方法（design method），这是在应对复杂的真实世界中的语法中产生的。复杂性在这里并不是指构式数量的庞大（虽然这也是很重要的一点），而是指要处理的相关语言现象的深度。计算机科学对于构建复杂系统有着丰富的经验，提出了一系列关于设计的概念，如设计模式（Design Pattern）、抽象（Computational Abstraction）、高级语言编译（Compilation from High Level Specification）等等。这些相同的概念对于语法设计（grammar design）来说都有很大的潜在价值。

一般地，FCG构式基于模板（templates）而定义。模板能描述常见的有关人类语言的设计模式，例如功能性的结构[[9]](#footnote-9)，一致性（agreement），场拓扑（field topology），价（valence），连接（linking）等等。不同的模板合在一起就能构建一个可操作的构式，使得模块化的设计变得可能，即便在操作层面[[10]](#footnote-10)上还无法清晰地表达这个模块化[[11]](#footnote-11)设计。（因为）便捷上的考虑应尽可能地与设计上的考虑分开。模板还在语言建模上扮演重要角色，因为它们可以作为基本的算子（primitive operators）来扩充或改变构式。

本文剩余篇幅将分别讨论三个层次的语法设计和实现：语言学层面介绍了构式语法中主要的语言学原则以及FCG中用到的语言学方法；设计层面则阐述了设计模式的相关概念，以及实现它们的模板；最后，操作层面介绍了FCG主要的表征机制和处理步骤，这一层面的介绍十分简略，更详细的内容将放在后续文章中。

**二、语言学层面**

FCG最原则性的一点是提倡功能学派而非形式学派的语言视角。

**2.1 功能学派和形式学派的语言视角**

功能学派（functionalist）和形式学派（formalist）的争论激烈且持久，尽管大多数争论的辩论色彩（rhetorical）要远大于实际意义。然而，有一点不同至关重要，它导致对一个人是如何处理和学习语言的截然不同的分析。在功能学派看来，语言的根本是交流的工具，说话者尽力在影响听话者脑中的认知活动，使听话者注意到真实世界中的某个具体方面，完成具体的动作，储存信息，等等。

在形式学派看来，语言研究的初衷并不是把它当作交际工具。语言分析的重点落在如何理解一个句子和其结构上。形式学派会把锤子描述成由一个圆柱体和一个（铁）块儿通过某种特定的绑定方式组成，而功能学派则会描述为一个手握的柄加一个锤击的头，其中柄和头通常[[12]](#footnote-12)具有圆柱体和（铁）块儿的形式。功能学派的视角强调对象的功能，主要是因为相同的功能通常能通过多种途径实现。例如，手柄可以插入锤头中，也可以被绑在锤头上；锤头既可以是一个（铁）块儿，也可以是另一个（木质）圆柱体。更激进地，一只鞋也可以是一个锤子，鞋帮是把手，鞋底是锤头。

在语言学中，持形式学派观点的人强调语言的句法结构，通常，他们会定义一个操作过程（procedure），能够生成语言中所有可能的、被认为符合语法的（judged to be grammartical）[[13]](#footnote-13)结构。而持功能学派观点的人更关注形式是如何获得（achieve）语法和语义功能的。如此一来，名词和动词的差异便不单单是结构性的——即他们可以填入哪种句法结构；而是功能性的——即他们在交际活动中可提供的功能：比如名词可以用作指称（nouns can be used as nominals）[[14]](#footnote-14)，以引入谈及的对象；而动词通常用作引入事件（event），以描述一件事情（affair）的状态。

一般来说，站在功能性的立场去设计和使用工具是有很多好处的。你对一个工具的每个组分的功能理解越深刻，你就越能正确地使用它；而一个功能性的视角也让你在特定的语境中更准确地找到一个工具的变体（variant），或者识别出一个并没有典型形状的工具，或者改进这个工具。显然，通常情况下人们并不会在全然理解一个工具之后才去使用它，而是去模仿他们看到的[[15]](#footnote-15)别人的使用方法。这是完全掌握其用法的第一步，且不可避免地要求一个功能学派的视角。

把语言看作工具能够获得同样的好处。功能性的分析将更好地理解为什么语言会是现在的样子，也使得语言理解和语言生产系统变得更灵活和鲁棒。此外，功能性的分析还提供了一种有别于纯统计性的语言学习技术，即使用功能性的推断来取代规约性（inductive）的推断[[16]](#footnote-16)。人们将通过重构（reconstruct）[[17]](#footnote-17)某个新引入的未知形式来推测它的意义和功能。

**2.2 语义功能和认知操作**

一旦我们接受了功能学派的视角，就很容易明白句子中的每个元素都有一个意义（meaning）和语义功能（semantic function）。意义，或曰语义内容（semantic content），是一个词代表的概念，而语义功能则是该构建块（building block）[[18]](#footnote-18)在解释（interpretation）过程中的操作。例如，“slow”在“the slow train”中会唤起与移动物体速度相关的“慢”的概念（意义），“slow”的语义功能，至少在这个短语中，是用来限定名词train可能的指称集合，特别是限定上下文中的慢速火车集合。

同一个意义可能有许多不同的语义功能，如下所示：

1. the slow train

2. they slow down

3. the slow go first

4. the train was slow

5. they ran slowly

所有这些句子都用到了“慢”的概念，它们不同在于它们要求听话者对这个概念的操作：1. 中用slow限制相关指称；2. 中对句子主语进行限定（circumscribe）；3. 中指称了一类物体；4. 中断言了主语的属性；5. 中对动词附加了一个属性。

语义功能这个概念通常只能模糊地定义。当我们想要形式化和操作化语法时我们需要一个更精确的定义。FCG假想了语义过程，即在解释句子时需要进行一系列对语义接地的世界模型和篇章模型（over a perceptually grounded world model and discourse model）的认知操作（cognitive operation）。每一个操作将产生一个可供其他操作使用的结果。这些操作包括集合操作、选取元素操作、筛选操作、对具体词条的知觉特征的计算操作、基于视角转换的物体位置的空间转换操作，等等。

以下是几个具体的认知操作例子：

1. get-context，从当前上下文中取出对象[[19]](#footnote-19)并将之存储于一个篇章模型（discourse model）中；

2. filter-set-class，从篇章模型中选取属于某个特定类别的对象；

……

6. describe-event，断言一个特定事件在当前上下文中发生

我们的团队研究出了一个意义表征系统叫“渐进补充语言”（Incremental Recruitment Language，IRL），可以用于接地的过程性语义（grounded procedural semantics）。IRL里的每一个认知操作都包含了一定的参数，其中一个（通常称为目标参数）是该认知操作的结果。认知操作可以通过互相利用结果结合在一起形成网络，如图2所示 [31] 。

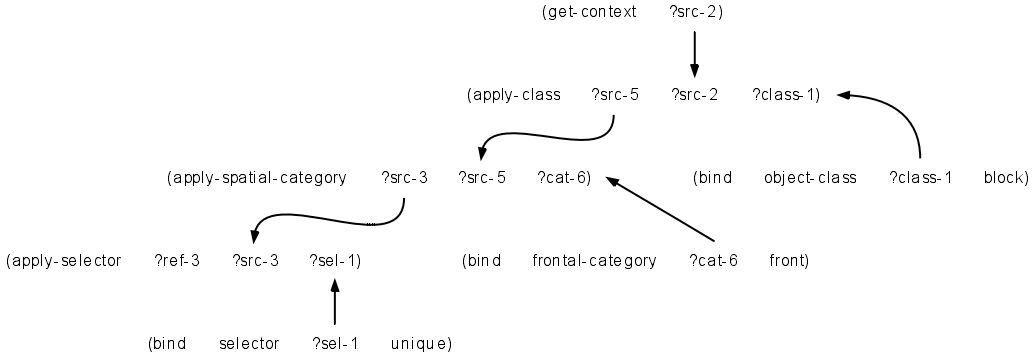


图2

图2表征了德语短语“der vordere Block”（the front block）的语义过程。当这样一个网络运行时，每一个认知操作根据真实世界模型和篇章上下文计算其（目标）参数值。这些值在网络中传递，直到没有新的计算为止。IRL在本书中并没有更深入的讨论，因为它的细节与语法处理无关。

译者注：上述网络可以写作

apply-selector (

apply-spatial-category (

apply-class (

get-context (),

bind (

object-class,

block //Block

)

),

bind (

frontal-category,

front //vordere

)

),

bind (

selector,

unique //the

)

)

我们现在可以更精确地讨论什么是语义功能[[20]](#footnote-20)。它指定了一个概念性的构建块在特定的认知操作中的角色（role）。例如，“slow”在“the slow train”中的语义功能是一个修饰语（qualifier），它在filter-set-qualifier操作中被赋予该功能。这个操作对篇章模型中的一系列对象进行筛选，筛出那些满足一个给定属性的对象。该操作的目标参数随后被用以限定相关指称。

与每个语义功能相匹配的还有一个句法功能，它与词类（lexical category）相联系。例如，“slowly”的语义功能是修饰表移动的“ran”，而其句法功能为副词性的[[21]](#footnote-21)。副词性功能的特点是以“-ly”结尾，把一个形容词转变为副词。

词类、句法功能和语义功能间的联系是双向的。见图3。

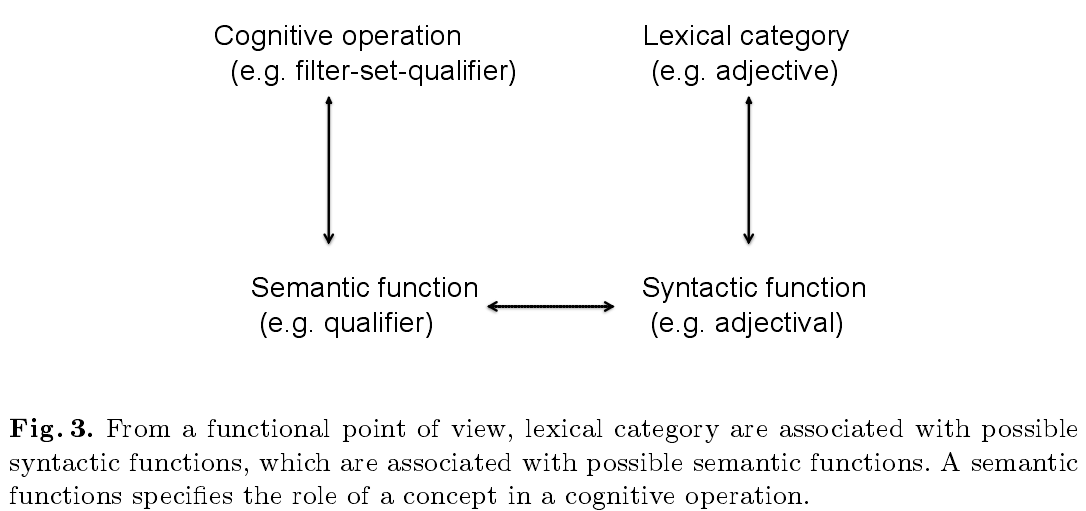


图3

词类提供的信息在语言理解过程中用来预测可能的句法功能，后者又可以预测语义功能。在语言生产过程中反之亦然。图4则对上述五个“slow”例句从这四个层次进行了描述。

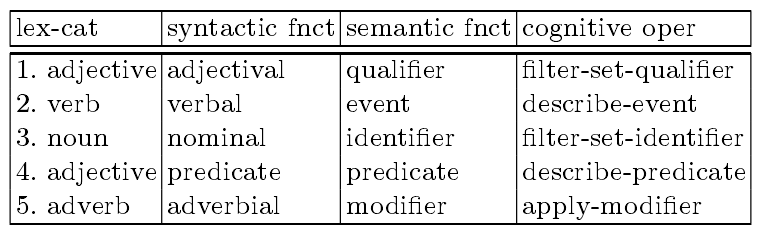


图4

**2.3 短语结构**

人类语言的一个重要特征是层次性或组合性。一个句子不仅是由单个词语组成的，还可以由同样具有句法和语义功能的词组组成。短语有特定的句法类型（如名词性短语），并能参与到更大的短语的句法或语义功能中去。一个短语的意义是由一个认知操作网络（network，以下简称网络）组成的。这些操作一般是由单词提供的，不过整个短语却可能附加额外的认知操作。

在功能学派看来，短语由其组分的功能定义。例如，一个名词性短语“the train”由一个定冠词和一个名词组成。名词的语义功能是在当前上下文中指定一个对象集合，而定冠词则进一步指定哪一个或一些元素需要被选取。

允许组分在短语中充当特定角色的约束（constraint）取决于语言。它通常包括以下几个方面：

1. 句法范畴：短语的组分通常需要有确定的词类或短语类型，至少具备参与具体句法功能的潜力，有可能是在强制转换之后。例如，冠词和数词通常只能作为名词短语的限定语。

2. 语义范畴：组分所指称的内容必须有确定的语义类型。例如，名词的意义能提供一个集合，且该集合被其所在名词短语限制。例如，一个复数的定冠词预示着它选择的对象必须是一个可数集。

3. 语序：短语的组分常常有语序限制。例如，组合性的颜色描述“light blue”要求非色彩范畴充当色彩范畴的修饰语。

4. 一致性：短语组分间通常有句法和语义的一致性约束。例如，法语中同一名词短语下的限定词、形容词和名词都受到性和数的约束。

短语构式的基本功能是定义以上特征并指定短语的属性（property），例如哪个特征从组分中渗透到整个短语里（例如法语中名词短语的名词如果是阴性的则整个短语也是阴性的），哪些附加的句法和语义属性属于整个短语，以及短语的意义是如何通过组合其组分的意义获得的。

**2.4 语法矩形**

词项和短语结构形成了句子的主干，因为它们展示（signal）了说话者想要传达的主要的概念性的构建块，也展示了这些构建块在解释句子时的作用。人类语言常常通过模块化并扩展基本结构骨架或增加词语的形态或语音变化来传达各种额外的意义（通常称为语法意义，grammatical meaning）。这与古典的人物画的基本结构类似，它先将（主要的）不同人物置于背景之上，再添加其他额外方面。例如，画家可能会使用一个特殊的调色给予景物某种感情色彩，或增加明暗区域的对比度以突出景物中的某些元素，或采用强烈的面部表情或手势增加戏剧效果。

在许多人类语言中表达的语法意义可以归入不同的语义功能域（semantic and functional domain）[[22]](#footnote-22)中。下面是几个例子：

论元结构：在基本短语结构之上，事件的参与者的角色可以用“who does what to whom”来更清晰地表述，例如使用“格”和词缀，或利用语序和前置词判定。

时体：现在/过去/将来时 v.s. 进行/完成/一般（repetitive）体。

态：虚拟语气，what evidence was available and how this was acquired, whether the source is reliable，等等。

限定（determination）：许多元可以更精确地描述（access）一个名词短语的所指，通常加上不同的冠词如“the”、“some”、“every”，或加上不可数、可数或有定、无定的形态标记。

地位（social status）：许多语言会通过加上形态标记或利用小品词，或选择不同的词项，表达说话者或听话者间的地位，简单的例子是法语中会用不同的词表达第二人称，如“tu”和“vous”。（译者注：就是尊称，俄语里也有类似的“ты”和“вы”；此外汉语、日语都有尊称）。

信息结构：许多语言能提供与篇章相关的信息结构，例如哪个对象是上文提及的，什么是主题等等。例如德语会把焦点性的主题提到句首或加上重音。

并不是所有的语法意义都能显式表达，在每一种语言中它们所表达和完成的都有显著的区别。不过，相同之处在于语法意义通过语义和句法范畴[[23]](#footnote-23)被被映射到句法形式上，如下图：

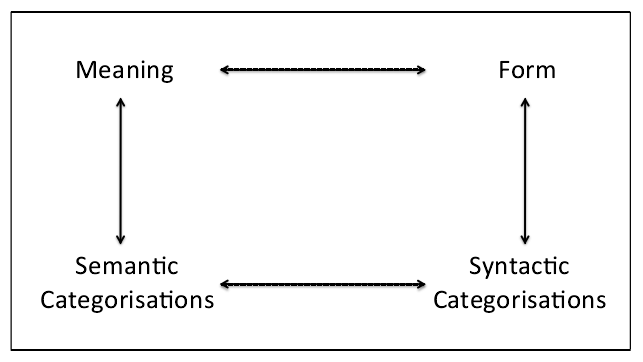


图5

从意义到形式的这种多层次特性是语法的特点，与词汇相反，后者直接把意义映射到形式上，而语法映射总是双向的。语言之所以采用间接的语法映射，而不是直接从意义映射到形式，是因为这样可以让意义和形式间的关系更加抽象，也因为还可能有其他因素参与到映射中，从而添加更多的信息。

下面是一个例子。在一个“推”事件中，推的角色（意义）并不是直接跟句法标记（形式）联系的，因为这对每一个动词来说都会有一个标记，标记集合会特别大。取而代之，参与事件的角色首先被范畴化为不同的语义角色，例如施事、受事等。这些语义角色又映射到句法的格，如主格、宾格等（如果没有格标记的语言则映射为主语、直接宾语等等），最后才映射到表层的标记如词缀或语序上。在每一步上，上下文，或额外的意义都会产生影响，例如在主动句中施事将映射到主语，但如果说话者想强调受事，它可以把受事放在主语位置，例如“the block was pushed to the edge of the table”。

语义范畴和句法范畴并不能总是绝对地区分开来。例如，许多语言把语义范畴表达为表层的形态，如英语的将来时（come – came – will come）。尽管如此，在分析语法时应该记住语法矩形的四个层面，并研究有哪些上下文因素影响了从一个层面到另一个层面的映射。

**2.5 分析步骤**

1. 第一步是确定语义功能域（delineate the primary semantic domain of interest and the communicative functions that will be investigated）。例如，某个研究可能关注空间语言，它用空间上的关系和视角去辨别场景中的对象，例如“the ball left of the box” [22] 。另一个研究可能关注对事件的描述，例如“the ball rolled off the table”，它可能隐含一个对事件的分类[[24]](#footnote-24)，并且要明确时间的体（完成体）。随后，相应的世界模型会从知觉和感觉过程中派生出来，而需要储存在篇章模型中的信息和语义域中使用到的概念也将被决定，例如空间范畴、事件类型、对象的图式（image schemas for objects）等等。最后，认知操作将作用于这些构建块和网络。

2. 下一步是调查哪些意义可以直接被词义覆盖。通常，词语会引入确定的概念，每一个词会有一个特别的潜力去表达语义功能。这个潜力来自于其所属的词类，词类将允许它具有一定的句法功能。

3. 第三步的焦点是短语，决定哪种短语可用于表达，每个短语是如何接入网络，要附加哪些额外的意义。这一步也将决定需要哪些句法和语义约束，例如，组分间需不需要一致性约束，哪些特征会从组分中渗入到短语中成为整体性的，等等。

4. 在基础的语义域之外可能还有其他语义域，表达语法意义例如时体态等等。对于这些语义域，分析者需要决定哪些需要在世界模型中表征，这些信息又如何从感觉运动中派生出来。这些域还将引入额外的认知操作和子网络，但通常不会引入新的句法功能。它们通过操作语义和句法特征。

**三、设计层面**

在上章中简介了FCG中的语言学基础。然而，将之转化为形式化的系统还有很大困难，因为语言在很大程度上并非模块化的，而且功能复杂。同一个元素（例如词）可以有许多功能，并能表达不同的语义域，而不同层次的约束（语用、语义、句法等）共同作用强烈。无论是在定义构式的复杂性（抓住所有约束）上还是在避免组合爆炸中都面临大的挑战。

FCG使用了计算机科学中的许多技术来处理复杂性。首先就是区分设计层面和操作层面；就像计算机中的机器语言和高级语言一样。在FCG中，操作层面是基于操作性构式（可应用于匹配matching和合并merging操作）的具体定义的。而在设计层面中，这些构式的定义则是基于模板的，相对会抽象许多。每一个模板会处理构式的一个特定方面并实现一个确定的设计模式。

**3.1 设计模式**

设计模式最初来源于建筑师。一个设计模式针对一个设计问题提出可复用的解决方案，该方案在具体应用场景中会有适当的调整。

在人类语言中，我们也找到了跨语言的通用设计模式。最熟悉的就是用短语表达复合意义。不同语言中决定哪些组分可以放在一起的机制差异很大，不过仍然有一些共同的机制。

另一个通用设计模式就是一致性。一致性意味着某些组分的句法或语义特征共享给了其他组分，它用来提示组分间的关系。

特征要么与单词绑定，要么就有明显的形态或语音标记。在短语中，它们常从某一组分渗入整个短语。例如，有定特征（definiteness）会从限定词渗入到限定性名词短语中。不同的语言会在使用一致性的场合、哪些特征需要一致性等方面存在不同，然而我们找到了一些通用的基础机制。例如，特征通常是成套的（bundle），一个特定的词或短语有多套特征。

另一个设计模式的例子是场拓扑（filed topology）。场拓扑是更灵活地表达短语语序的方法。在多数语言中，用meets-关系就能描述语序，meets (X, Y)意味着X紧跟Y出现。不过有些时候这个表征太弱了，例如在德语中几乎所有组分都能置于句首作为强调的话题。

场拓扑给每个短语绑定了一些“场”。一个场能够获得多个组分，从而最终的语序取决于顺序遍历每个场内的填充物。例如，德语中假设了五个场：前场（front field），能够获得话题；左界（left bracket），通常填入主要动词；剩下的三个场分别是中场（middle field），右界（right bracket，通常填入非主要动词）和后场（end field） [24] 。

场拓扑最初只是用在德语和荷兰语中，但它还适用于更多语序现象中。

一致性约束和场拓扑是面向句法构建和形态标记的设计模式。此外还有其他面向语义方面的设计模式。例如，短语中由单词形成的网络，和不同短语间形成的网络。这种设计模式称为连接模式（linking pattern）。

**3.2 构式**

构式是FCG操作层面的基本单元。它的语义端（Semantic Pole）包含语用、意义、语义结构和语义范畴信息，而句法端（Syntactic Pole）包含句法结构、句法范畴及语音和形态标记信息。构式一般有一些代表其组分的单元（Unit），它们用特征（Feature）来表征信息。

在语言理解和语言生产过程中构建的数据结构称为临时结构（transient structure）。它与构式一样由单元和特征组成，也分为语义端和句法端。最初，临时结构只有一个单元（称为top-unit），包含了在处理过程开始时一切已知的信息。在语言理解过程中，它包含了一个form特征，记录了整个表达的形式特征；在语言生产过程中，它包含了一个meaning特征，记录了说话者想要表达的完整意义。

构式被看作意义和形式间的双向绑定。在语言生产过程中构式的语义端去匹配（Match）临时结构的语义端，以检验该构式是否能应用上，（若匹配成功）进而，构式的句法端并入（Merge）临时结构的句法端。在语言理解的过程中，反之亦然。一个构式构式因此为另一个构式的应用进行了铺垫（prepares the ground），我们最终得到了一个构式应用链。（图7）通常，能够应用上的构式不止一个，我们因此得到了一个有着多条搜索路径的搜索空间，并要从中找出最优路径。

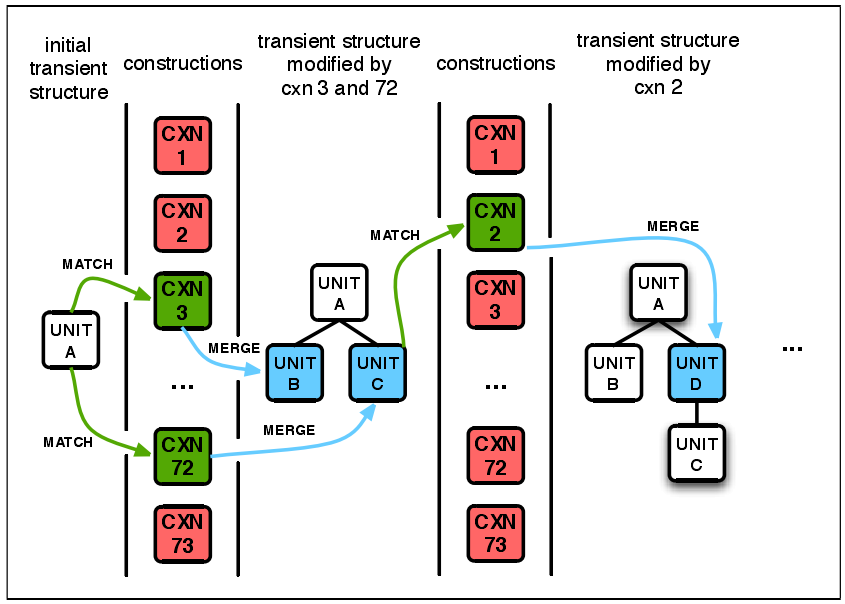


图7

构式是复杂的，因为其具有双向属性。对于限定性名词短语（determiner-nominal phrase），光说它由一个限定词和一个名词组成是不够的，我们还要说我们什么时候会用到这么一个短语（在语言生产过程中），以及限定词和名词的意义是如何组合的（在语言处理过程中）。此外，我们还要指定一致性约束、语义渗透，以及网络间的连接。构式复杂性的另一个原因是句法和语义必须同时考虑。一个构式在构建结构之前应该包含足够多的约束以最大化避免搜索。

FCG把构式分成不同的几类：

1. 词汇构式，只引入词干，它们指定意义、语音、句法和语义范畴，和词形；

2. 形态构式，引入词缀。它们指定词缀可以与何种词汇结合、结合的形式、句法和语义范畴、形态形式、及语音特征；

3. 功能性构式，用来定义词类、句法功能和语义功能间的联系；

4. 短语构式，用来指定短语上的约束，比如应该包含什么组分、有着怎样的一致性和语序等等。

**3.3 模板**

设计模式并不直接转换为构式，因为一个构式需要整合多个设计模式，而一个设计模式也可以对多个构式产生影响。例如，实现“一致性”这个设计模式就需要词汇构式、形态构式和短语构式引入句法和语义特征。即便如此，还是有办法去把设计模式的一些基本面进行抽象、并隐藏掉实现细节，这就是模板的功能。

一个模板决定了一个构式的某些方面。它有很多槽接受参数来决定它究竟如何实例化。当一个模板应用于一个构式时它扩展了这个构式（图8）。

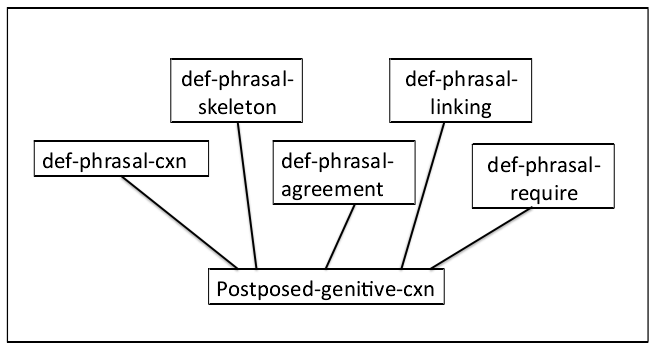


图8

模板是一个开放集，语法设计者可以根据他们感兴趣的方面设计模板。FCG的每一个发布版本都会提供模板库。一个模板越特定，它就越容易聚焦语言中的特定方面，但也越难移植到另一门语言。

**四、操作层面**

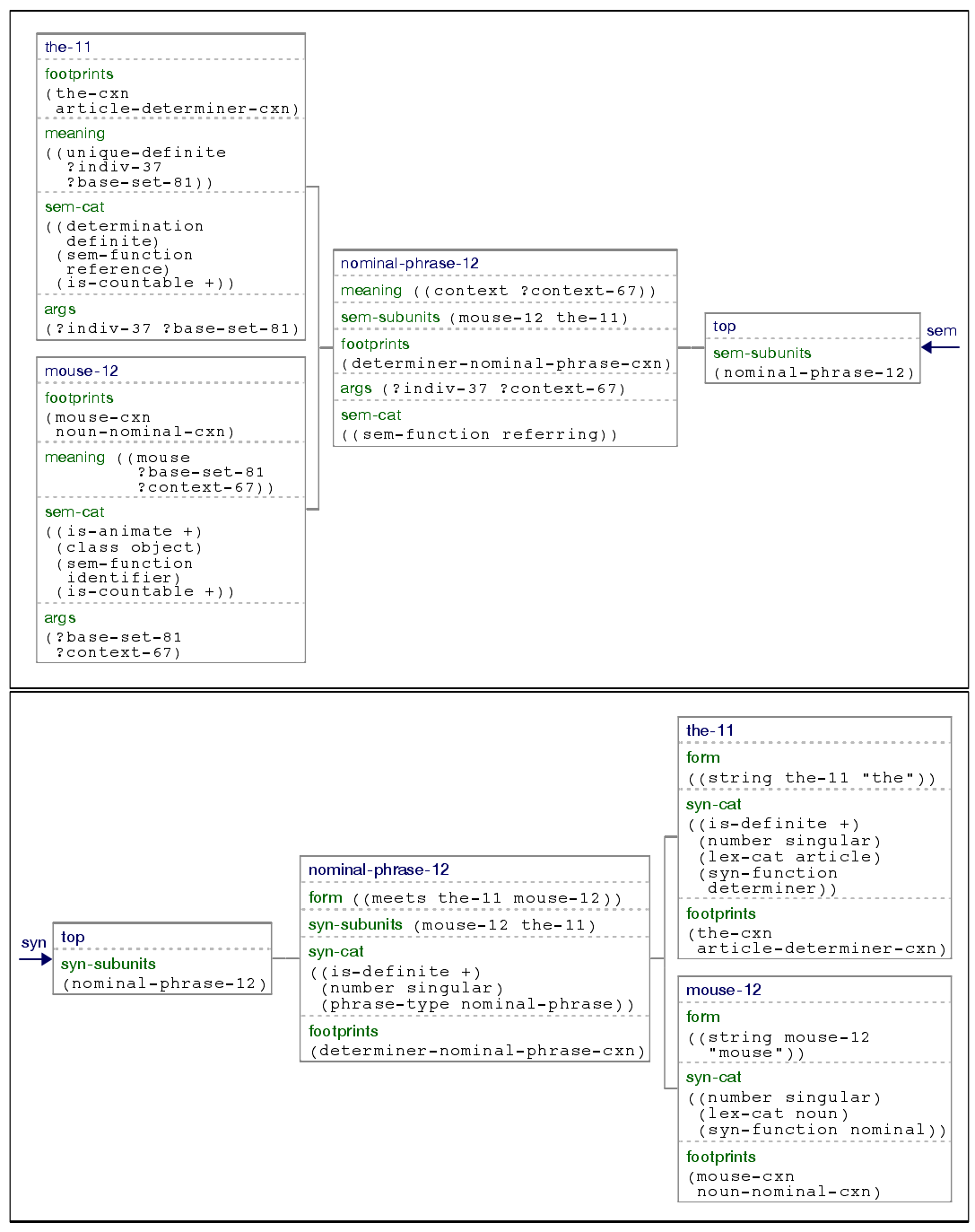
本文提倡自顶向下的设计，从语言学层面进行分析，明确语义域和语义功能，及需要表达的交际功能；然后再是设计层面，找到合适的设计模式和模板。

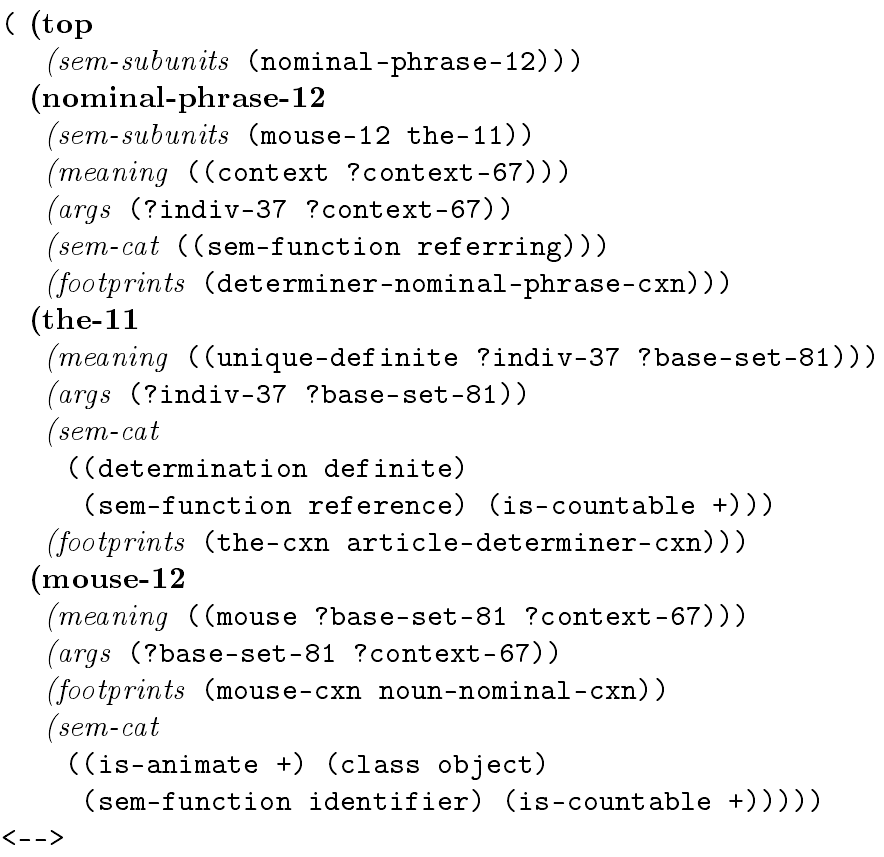
我们现在来到了操作层面，即定义一个构式和其所有细节。虽然构式可以徒手定义，但用模板会更加简单。本章接下来的部分将简介操作层面的主要结构和操作。

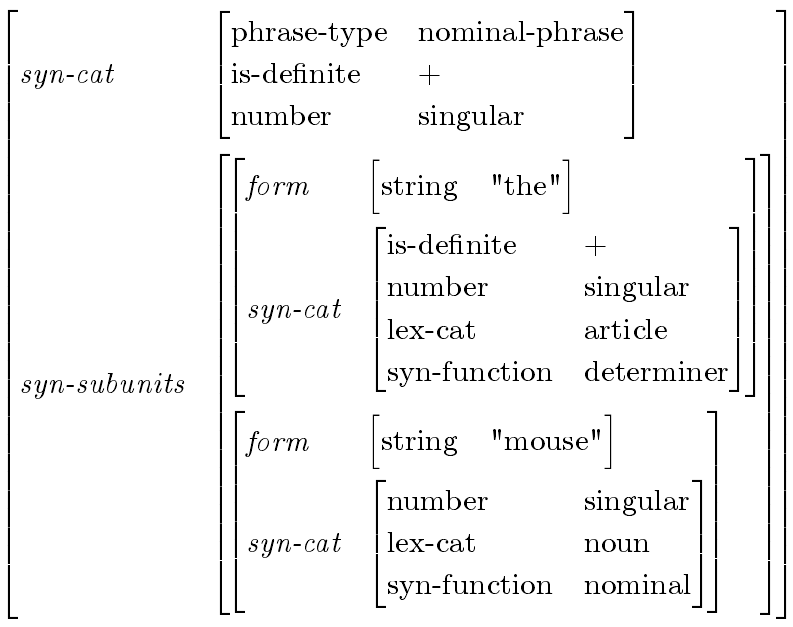
**4.1 表征临时结构**

FCG使用特征结构（feature structure）来表征语言理解和生产过程中产生的信息，这也遵循了当下计算语言学的主流方法。在语言理解中，临时结构一开始包含了所有能从表达中得到的信息（字符串、语序、语音信息）并逐步重构（reconstruct）语义和句法结构，以及意义。在语言生产中，临时结构一开始只包含了所有待表达的意义。不同的构式逐步覆盖这些意义，并构建起短语结构、约束句子形式知道完整的句子被派生出来。

特征结构用包含特征（Feature）和值（Value）的单元（Unit）来组合语言结构。单元有名字，这些名字可以与构式内用于引用的变量绑定，因此，层次结构可以显式地用一个特征subunits表征，它的值为所有子单元。







**4.2 表征构式**

构式与临时结构的表征是一致的，它们由包含特征和值的单元组成。构式比特征结构更加抽象。它们留出了足够的信息空白，所以它们能够匹配很多临时结构。它们包含变量可以与临时结构中的特定值绑定。它们也会用特殊的算子来指定值的一部分，例如包含算子（只要某些元素出现即可）、唯一算子（特定的元素只能出现一次）、排除算子（特定的元素不能出现）。

**4.3 过程注记（Procedural Annotation）**

过程注记由构式提供的额外信息组成，用以执行构建操作，或避免构式持续不断地应用。构建结构需要两个操作：一个用来创建新的单元并将之挂靠在一个已有单元之下；一个用来向新单元中填入信息。

**五、结语**

本文介绍了一些FCG的设计原则。区分了分析的三个层次。

FCG是对构式语法中漂浮的许多构想的一次实践。例如，继承（inheritance）在许多构式语法中扮演了重要角色，但未被FCG采用。虽然FCG采用了许多与其他基于合一运算的语法相似的技术，但还是存在不同，例如FCG把语法和语义分作两端。

**总结部分**

本文的第三、四章的所有内容均可以在“2015\_01\_spring/report/\*田骏”路径下找到更详细的解释。这里只做简略介绍，为了使得“语言层面-设计层面-操作层面”三层介绍显得完整。

本文最核心的部分显然是第二章有关语言学层面的讨论。而第二章中最核心的部分显然是两个矩形：在词类和认知操作中起联系作用的句法和语义功能（图3），和形式、意义间其联系作用的句法和语义范畴。这两个矩形是对形、义结合的操作化描述（设计），体现了FCG的特点。

正如作者在设计层面谈到的，每一个设计模式都是针对一个具体的问题而提出的，FCG的设计是出于机器人实现语义接地性质的交流（即要描述、表达客观实体，语言要落在实处），因此在2.5节中作者指出，最开始应该先确认所有的可能参与到交流活动（又称交流游戏）的对象，它们组成了世界模型（可参考我在15年春第二次报告中的《Grounding Language》翻译）。然后再依次把意义翻译成形式或把形式解读成意义。

**评述部分**

**一、形式学派视角与功能学派视角**

形式学派攻击功能学派最主要的一点是开天窗说大话，不落在实处。功能学派反击的理由是你能够说清楚所有“实处”吗？言下之意是“太落在实处了”。

比如针对“而一个功能性的视角也让你在特定的语境中更准确地找到一个工具的变体（variant），或者识别出一个并没有典型形状的工具，或者改进这个工具”，形式学派会认为，没有典型形状不代表没有形状，对于每一种形状都应该描述清楚，否则如何“准确地”找到变体？而功能学派则认为，变体不是你想找，想找就能找。“把鞋当作锤子”不是一个稳定的变体，而是一种临时性的、不可预见的行为。但这种行为不是不可捉摸，而是有一种更深层次的原因（鞋与锤子具有某种相似性），这是一种规律的规律、语法的语法——这就是功能的本质。FCG其实应该好好地突出它的修复（Repair）过程，因为这才是形式学派所关心的。

形式学派视角不如功能学派视角好用的另一个实处在于“翻译”。比如，用形式方法，如何翻译2.4节中的“backbone”和“skeletal”？

Lexical items and phrase structure form the backbone of a sentence …

… by modulating and extending the basic skeletal structure …

又比如，为什么有的文献喜欢用domain，有的却又喜欢用scope？虽然二者从词源说确实有细微的不同，但在今天，至少在计算机术语里，这两个词是同义的。Nevertheless，在碰到它们的时候人们还是会喜欢分别翻译为“定义域”和“作用域”。而如果再加上一个定语，如semantic，简直没法再翻译下去。

“直译”代表的是形式方法，而“意译”更多的是功能方法。为什么要意译呢？并不是直译不对，而是在不同语言中，人们表达一个概念的方式是不一样的。汉语喜欢说“壮如牛”，英语则说“as strong as a horse”，这是为什么商业领域或学术领域很多人喜欢时不时夹带一句英文——并不是他们爱显摆，而是这概念就是从英文中来的，必须用英文表达才准确。

所以生成语法的Internal-language对于解决翻译问题而言是远远不够的，除非它能深入概念层面。

**二、FCG的形式化思想**

这篇文章中最费解的一句话要数“此外，功能性的分析还提供了一种有别于纯统计性的语言学习技术，即使用功能性的推断来取代规约性（inductive）的推断。人们将通过重构某个新引入的未知形式来推测它的意义和功能。”

“纯统计性”不知道是在说形式语言学，还是机器学习，甚至是基于使用的观点派。不过至少形式语言学本身是不承认自己是“纯统计性”的。

另一方面，所谓“重构”，并不是很高深的技术，只是把你接受到的一个线性句法形式分析为一个意义结构，“重新”是相对于说话者心中的意义结构来说的，并不是“创新”。

再有就是，FCG是承认组合性的，这好像也比较符合形式学派的观点。也更加模糊了FCG的立场。

我们的看法是，一旦一种方法开始试图加入规则，或曰指导（supervisor），它就不可避免地与形式化沾边。至于有没有“流于形式”，成为“形式主义”，那就是各家之言了。FCG认为自己不是形式主义，但它的设计模式、包括运算法则（合一运算），都有形式主义的影子。它排斥的形式主义，是“低级的形式主义”，是句法先行的形式主义，而它实际上是比较高级的形式主义。因为它所有的构式，都是脱离了具体实例存在的。这就是一种形式主义。准确说，是“把功能形式化了的形式主义”。

我们认为FCG提出的某些“低级的形式主义”的缺点是正确的，但它的做法并没有显著地摆脱这种缺点。它仍然依赖人们赋予它各种“高级的”规则，而不是自己从语言事实中进行提炼。所以其实我们对FCG的形式化操作并不是很感兴趣。

**三、对毕业论文的贡献**

FCG给我们最大的启示在于它的功能主张：即语言是善变的、用法是多样的，不要老拘泥于对形式的讨论，而要思考到底是什么决定了现在的形式。

对应到四字表达，就是不要只考虑格式，而要考虑格式是怎么来的。上面总结的“语法的语法”，就是我们要研究的“能产方式”。能产方式只有套格式一种吗？那格式是怎么来的呢？因此我们提出，人们真正地花在“套格式”上的心思是很少的，大多数时候人们是通过“仿造”来构建新的表达的；也有时候人们希望强调，而把双字词扩展为四字表达；还可能人们因为误用或为了吸引注意力而变换了语序。格式只不过是人们总结出的形式规律，并不能完全指导人们的行为。

1. Association在文中出现了很多次，不过在与bi-directional组合时倾向于翻译为“绑定”。双向绑定是一个设计模式术语。例如“上传下达”是单向的，再加上“虚心纳谏”才是双向的。双向绑定侧重于强调两个端的呼应。 [↑](#footnote-ref-1)
2. 意为只用一套语法就能处理两个方向的过程。 [↑](#footnote-ref-2)
3. 这里的模块主要指语法、语义、语用、形态和语音五个语言学模块。 [↑](#footnote-ref-3)
4. 语法grammar并不是句法syntax，它是与词汇lexicon相对的、泛指词汇间关系的概念。 [↑](#footnote-ref-4)
5. 这里说明了“额外”并不是必须的。 [↑](#footnote-ref-5)
6. Derive，这里其实翻译为“产生”贴切。但“产生”还有许多其他的表述。为了不至于混淆，这里采用直译的方法。 [↑](#footnote-ref-6)
7. World model，指针对某个具体场景而建立的对该场景内事物的全部感觉认知，包括背景知识和即将发生的上下文。 [↑](#footnote-ref-7)
8. 这里的语法grammatical并不是与语义相对的句法，而是包含语义的上位概念。 [↑](#footnote-ref-8)
9. functional structure，不是一个概念，例如句法结构、语义结构都算是功能性结构。 [↑](#footnote-ref-9)
10. 模板是设计层面的概念，无法应用到操作层面（底层）。 [↑](#footnote-ref-10)
11. 这里的模块化设计与上文否定的“分模块”不冲突，二者都表达要综合考虑多个模块的意义，但二者的模块所指不一样。 [↑](#footnote-ref-11)
12. 通常的意思就是并非一定。 [↑](#footnote-ref-12)
13. 这句话的潜在意义是“语法是主观的，而形式学派会[有意或无意地]选取那些有利于形式完备的语言现象，而忽略不利的现象”。 [↑](#footnote-ref-13)
14. nom-, nym- 都是name的变体。 [↑](#footnote-ref-14)
15. 这里加上了一个限制“他们看到的”，意思是说，“别人的语法是不是普遍的语法我不管，我只能模仿我所能看到的别人的语法”。 [↑](#footnote-ref-15)
16. 这里没有细说，但体现出的还是理性主义的思想。是比一般认知语法更高一层的思想。一般认知语法只会说“某个表达统计性地先入了另一个表达”，但作者却要问“为什么会发生先入”。 [↑](#footnote-ref-16)
17. 重构并不是修复（repair），而是把一个线性的表达重构成层次性的表达。有点哗众取宠的意思。 [↑](#footnote-ref-17)
18. 构建块指的是元素。 [↑](#footnote-ref-18)
19. 对象（object）与元素（element）在文中是两个不同的概念，其关系应该类似于type和token。 [↑](#footnote-ref-19)
20. 重新回顾一下，语义功能（semantic function）与意义（meaning）相对，后者又称为语义内容（semantic content）。 [↑](#footnote-ref-20)
21. 这其实就是把我们的“粗略”的分析精细化了。我们认为“ran slowly”中，slowly作副词，修饰动词ran。而作者认为，这是四个描述：slowly的意义是“慢”，它的语义功能的是修饰ran，句法功能是作副词（副词的句法功能是修饰动词），而副词一般出现在动词后面。 [↑](#footnote-ref-21)
22. 域就是定义和作用的范围，等价于scope。 [↑](#footnote-ref-22)
23. 回顾一下，语义和句法范畴是一种约束（constraint），约束是在短语构式中定义的，这是与语义和句法功能不一致的地方，后者是在词构式中定义的。 [↑](#footnote-ref-23)
24. 例如是滚下去的不是滑下去的 [↑](#footnote-ref-24)