第 一 编

自然语言逻辑的基础 —— 形式语义学

形式语义学(Formal Semantics)也叫逻辑语义学(Logical Semantics) 其基本特征为：运用真值条件模型论的方法对语言进行语义解释，即以数学结构方式表现出来的外部世界的模型作为参照物来确立语句的真值条件意义。一般来讲，形式语义学的概念不限于自然语言的领域，但作为自然语言逻辑的基础，这里主要指自然语言的形式语义学。从互相关联的角度来说，形式语义学还包括形式语形（句法）学，因为要对自然语言进行形式语义学解释必先对此进行形式句法学的处理。若对自然语言的语义解释进行延伸，势必进入语用学的领域，所以，本编所谓自然语言形式语义学的概念还适当附带一些形式语用学的内容。其次，自然语言的形式语义学，一方面关注自然语言与逻辑人工语言的共同之处，用处理逻辑语言的惯用方式来刻画自然语言的语义；另一方面，形式语义学则更为关注自然语言的特殊情况，有时据此还反过来对所用的逻辑方法不断进行改进，以至演绎出从蒙太格语法(Montague Grammar)开始到先后涌现出的广义量词理论 (Generalized Quantifier Theory）、话语表现理论(Discourse Representation Theory)、情境语义学(Situation Semantics)以及类型-逻辑语法(Type-Logical Grammar) 等一系列形式语义学流派的发展过程。

需要指出的是，自然语言的形式语义学研究显示出理论与实践的双重价值。如广义量词理论的研究要回答所有自然语言有关名词短语或限定词的语义共性(semantic universals) 的问题，要证明限定词在总体上表达力如何？有多大范围？自然语言常用的限定词不过几十个，而广义量词理论的研究表明，在一定论域条件下，与限定词对应的二元量词的可能数目竟有六万多个，就是那些与自然语言联系最紧密的具有所谓“ conservativity ”性质的量词在理论上讲也有 512个，这些毕竟是非形式研究难以回答的问题；在当今语言的信息处理实践活动中，形式语义学关于自然语言的研究成果常常被计算机人工智能领域的学者所关注。电脑对自然语言的理解和分析必须是形式化的，“如果形式语义学的

分析采取计算机程序的方式，很显然在此意义上形式语义学者就变成了计算语言学家（参见 M.King : Epilogue: on the Relation between Computational Linguistics and Formal Semantics. In M.Rosner and R. Johnson ed. (1992): *Computational Linguistics and Formal Semantics*. p.284, Cambridge University Press)。

第一章自然语言形式语义学的研究方法

第 1 节关于自然语言句法的数理方法

1.1 自然语言的逻辑结构层次

在现代逻辑看来，自然语言跟逻辑人工语言一样，也是一个符号体系。自然语言体系内用词组句，由较小的语言单位形成较大的语言单位的现象就是符号之间的排列组合。自然语言各类表达式，尤其是句子表达式的深层部分实际上就是一种逻辑结构。由于现代逻辑工具有简繁之分，或者说不同的逻辑工具有不同程度的表达能力，所以，对自然语言表达式逻辑结构的分析就存在着程度深浅的差异。当然，这种差异不是相互抵触的，而是体现出一种分析层次的逐步深入。换言之，对同样的自然语言表达式，若用较简单的逻辑工具进行分析，得到其较简单的逻辑结构，若改用较复杂的逻辑工具，就会获得其更深刻的逻辑结构分析。如对英语句：

(1) John walked rapidly and Bill ran slowly.

命题逻辑仅仅关注(1)中“ and ”这个词的逻辑特性,用逻辑合取词“ & ”来表示它。命题逻辑并不考虑“ and ”左右两部分各自不同的内部结构，只是把它们看作是两个不可分的整体，即两个不同的命题，分别用 ϕ与 ψ表示之。所以，在命题逻辑看来，(1)的逻辑结构就是：

$$\phi \& \psi$$

而一阶谓词逻辑考虑的因素则更多一些，对自然语言句子的逻辑结构分析更深入一些，对（1)中“ and”左右两边的句子进一步考察其内部构造，区分出其中的个体词与谓词。 “John ”与“ Bill”等专名指称个体，分别用个体常项 j 与 b 来表示。而“ walked rapidly ”与“ran slowly ”分别对应谓词“ walk-rapidly ”与“ run-slowly ”，[1]这里不考虑动词的时间因素以及动词短语的内部结构。于是，在一阶谓词逻辑看来，(1)的逻辑结构为：

walk-rapidly (j) & **run-slowly** (b)

高阶谓词逻辑的分析则更精细一点，对(1)中的不及物动词短语的内部结构予以关注。高阶逻辑认为“walk”指称的是个体的集合，其类型为⟨e, t⟩，“rapidly’，对应更高层次的函词，指称从个体集合到个体集合的函项，其类型为⟨⟨e, t⟩, ⟨e, t⟩⟩ 。于是(1)的逻辑结构就表现为：

(**rapidly** (**walk**)) (j) & (**slowly** (**run**)) (b)

这里“ rapidly ”与“ walk ”组合的直观理解为：“walk ”对应的个体集合作为“ rapidly”对应函项的主目，其运算结果产生另一个体集合，这个集合是在“walk”对应集合的基础上进行限制而得，即是“ 走得快，的个体集合。[2]高阶逻辑的分析仍没有考虑(1)的时间因素，蒙太格语法中的内涵逻辑兼有时态逻辑的表达力，补上这个缺欠，对(1)进行更深入分析，其逻辑结构就是：

PAST[(**rapidly** (**walk**)) (j)] & **PAST**[(**slowly** (**run**)) (b)]

[1] 按惯例，用英文词的黑体来表示与之相应的逻辑符号。

[2] 而不是“ 走 ”的个体集合与“ 快 ”的个体集合的交集。

这里“ PAST ”为表过去时间的时态算子，其直观理解为：PAST 后面的公式在说话时间（或现在时间点）之前的某个时间点参照下为真。

1.2 λ-表达式的运用

在现代逻辑中，λ-算子是刻画函项的工具，λ-表达式对自然语言表达式逻辑结构的描述起到一种不可替代的绝妙作用。从自然语言逻辑的符号学角度看，自然语言的句法语形的生成实际上就是一种符号的组合。而包含λ-表达式的λ-演算系统本质上是一种组合逻辑，所以，自然语言逻辑在处理自然语言表达式句法生成时，就大量地运用了λ-表达式及其演算的方法。形式语义学的重要特征之一是贯彻组合原则于自然语言的句法与语义等各个方面。就自然语言的句法方面而言，英语句“ John walks”是由其部分“ John ”与“ walks ”组合而成。就自然语言的逻辑结构而言，翻译或表达“ John walks ”的逻辑式是由翻译或表达“ John ”的逻辑式与翻译或表达“ walks ”的逻辑表达式组合而成。翻译“ John ”或“ walks ”的逻辑表达式是什么样的？这就要考虑到组合的需要。用λ-表达式来表现对英语词的翻译，是有利于贯彻组合原则的。例如，把“ John ”翻译成“ λPP(j) ”模样的λ-表达式，把“ walk ”翻译成“ λx **walk**(x) ”模样的λ-表达式。则“ John walks ”的翻译式就是“ John ”的翻译式与“ walks ”的翻译式的组合，即“ λPP(j)[λ x **walk**(x)] ”。进行λ-转换得：[λx **walk**(x)](j).再次转换得： **walk**(j)。这就是翻译英语句“ John walks ”所获得的逻辑结构。这里要问为什么事先知道把“ John ”翻译成“ λPP(j) ”而把“ walks ”翻译成“ λx **walk**(x) ”模样的λ-表达式。这涉及运用λ-表达式的演算方法，涉及一种“倒推法”。根据传统理解，英语句“ John walks ”的逻辑结构翻译应该是 **walk**(j)。以此为出发点，利用λ-演算的方法，步步回溯，最终确

立“ John ”与“ walks ”的λ-表达式。其过程为：

walk (j)	(“John walks ’的翻译)
[λx **walk** (x)] (j)	(据λ-逆转换)
ιPP(j)[λx **walk**(x)]	(据λ-逆转换)
所以λPP(j)是 John 的翻译，	(据组合原则，“John walks ”
λx **walk**(x)是 walks 的翻译	翻译是“ John’,的翻译与
	“walks ’的翻译的组合)

λ-表达式在形式语义学对自然语言的处理中起到的独特作用，还表现在对自然语言某些虚词的句法功能意义的刻画。如英语句：

(2) John has slept since midnight

其逻辑结构的展现关键在于介词“ since ’的意义描述。在 Dowty 系统中，“ since ”被翻译成λ-表达式：

$$\lambda\mathscr{P}_t\lambda P_t\ \mathscr{P}_t\{\hat{}\lambda t_1[\forall t_2[[t_1 < t_2 \,\&\, XN(t_2)] \rightarrow P_t\{t_2\}]]\}$$

据此可把(2)翻译成：

$$\forall t_2\,[[\,\mathbf{midnight} < t_2 \,\&\, XN(t_2)\,] \rightarrow AT(t_2, \mathbf{sleep}(j))]$$

其直观理解为：对任一时间区间 t_2而言，若 t_2在半夜之后并且延续到现在，则在 t_2这段时间内 j 所指个体睡觉。这就比较恰当地显示出(2)的语义特点。

1.3 递归定义方法

自然语言是一个符号系统，自然语言的生成就是符号串的组合。怎样刻画符号串的组合？除了用 λ-表达式刻画自然语言基本语词便于组合外，对自然语言复合词组乃至句子的生成，形式语义学还采用递归定义的方式来获得。这里递归定义的运用涉及两方面的因素：一是字母表，也即初始符号的集合，记为 C ；二是定义的内容。例如， $C = \{a, b\}$，据此可以递归定义一个由 C 中元素组合成的符号串的集合 M 如下：

(i) $aa \in M \ \& \ bb \in M$;

(ii) $\forall x\,[x \in M \rightarrow axa \in M \ \& \ bxb \in M]$;

(iii) M 除包含由 (i)与(ii) 生成的符号串外，别无其他。

(i)称作递归定义的基础，即用递归定义生成符号串的出发点。(ii)是所谓递归定义的步骤，意为 若符号串 x 属于 M ，则 在 x 两端各毗连一个 a (或 b)所得仍属于 M 。(iii)是定义的限制，排除那些并不是由(i)或(ii)生成的符号串，如 ab, aaab 等。用递归定义生成的符号串还可用推演的方式来证明生成是合法的，如证明 baab 是 M 中生成的符号串：

(a)	$aa \in M \ \& \ bb \in M$	(M 的定义(i))
(b)	$\forall x\,[x \in M \rightarrow axa \in M \ \& \ bxb \in M]$	(M 的定义(ii))
(c)	$aa \in M$	((a)[& -])
(d)	$aa \in M \rightarrow aaaa \in M \ \& \ baab \in M$	((b)[$\forall$ -])
(e)	$aaaa \in M \ \& \ baab \in M$	((c)(d)[$\rightarrow$ -])
(f)	$baab \in M$	((e)[& -])

在现代逻辑那里，递归定义方法的运用是非常广泛的。命题逻辑合式公式的形成也用递归定义表示。设字母表：

$$C = \{p, q, r, \sim, \&, \vee, \rightarrow, \leftrightarrow, (,)\}$$

合式公式的集合记为 WFF，则有

(i) $p, q, r \in WFF$ ；

(ii)对所有 α, β 来说，若 $\alpha, \beta \in WFF$，则：

(a) $\sim\alpha \in WFF$ ；

(b) $(\alpha \& \beta) \in WFF$ ；

(c) $(\alpha \vee \beta) \in WFF$；

(d) $(\alpha \rightarrow \beta) \in WFF$；

(e) $(\alpha \leftrightarrow \beta) \in WFF$ ；

(iii) WFF 中除由(i)与(ii) 生成的符号串外，别无其他。

判定$((p\&q)\vee r)$是 WFF 中合式公式的推证如下：

(a)	$p \in WFF \& q \in WFF$	(WFF 的定义(i))
(b)	$p \in WFF \& q \in WFF \rightarrow (p\&q) \in WFF$	(WFF 的定义(ii))
(c)	$(p\&q) \in WFF$	((a),(b)[$\rightarrow_-$])
(d)	$r \in WFF$	(WFF 的定义(i))
(e)	$((p\&q) \in WFF \& r \in WFF \rightarrow ((p\&q)\vee r) \in WFF$	(WFF 的定义(ii))
(f)	$(p\&q) \in WFF \& r \in WFF$	((c), (d) [$\&_+$])
(g)	$((p\&q)\vee r) \in WFF$	((e), (f) [$\rightarrow_-$])

禀承逻辑的传统，形式语义学研究自然语言也大量采用递归定义的方法。尤其在自然语言的语句系统中，形式语义学用递归

(i) 对每一 $\alpha \in X_T \cup X_{IV} \cup X_t$ 来说都有， $\alpha \in A$ ；

(ii) 若 $\alpha, \beta \in A$，则 $F_0(\alpha, \beta) \in A$ ，这里

$$F_0(\alpha, \beta) = \begin{cases} \alpha\beta & \text{若 } \alpha \in X_T, \beta \in X_{IV} \\ \varnothing & \text{否则} \end{cases}$$

(iii) A 中除由 (i)与(ii) 生成的符号串以外，别无其他。

运用(i)与(ii)生成的表达式其中有： John walks, John talks, Bill walks 及 Bill talks 。它们作成的集合(A 的子集)便是该微型系统所生成的自然语言的句子集合。蒙太格语法中的自然语言系统大都采取上述进行适当限制的递归定义来生成自然语言表达式。此外，范畴语法中所谓斯蒂德曼 (Steedman)树形图的定义则更是完全采用递归的方式给出：

这里令初始符号的集合为 BasExp

(i) 对 任 $e \in BasExp, e \in Tree$ ；

(ii) 若 $T_1, \cdots, T_n \in Tree$ 则 $\langle C, \langle T_1, \cdots, T_n \rangle\rangle \in Tree$ ；

(iii) Tree 中除由(i)与(ii)生成以外，别无其他。

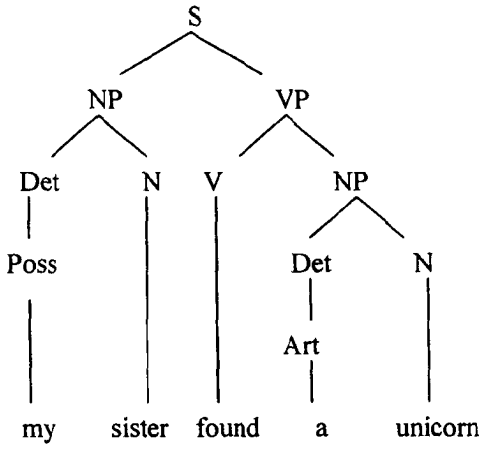
这里 BasExp 指英语词条的集合， C 指与英语表达式相应的范畴标记及其 λ-词项的序对， Tree 的成员根据(i)与(ii)一步步生成。详细情况请见下文第 14 页的介绍。

1.4 树形图方法

在形式语义学中，对自然语言表达式的句法结构生成，常用树形图的方式表达。树形图(tree diagram)又叫成分结构树(constituent structure tree)。这在转换生成语法等形式语言学流派那里也

用得极为普遍。如：

(3)



这样的树形图表达了关于自然语言句子的句法结构的三方面的信息：

(i) 句子的组成成分有等级层次的分类；
(ii) 每种成分的语法类型；
(iii) 这些成分从左到右的顺序。

在(3)中，由 S 标记的成分是最大的成分，它由成分 NP 与 VP 构成。而 NP 又由成分 Det 与 N 构成，等等。在(3)中， NP 在 VP 之前，Det 在 N 之前，等等。树形图由结点(node)组成。每个结点配上一个语法范畴(S, NP, VP, …)或词素(my, sister, …)作为标记(label)。连接结点之间的线段称为树枝(branch)，树枝总是连接一个较高的结点到一个较低的结点。

树形图的结构涉及几个概念：支配 dominance) 、居前(precedence)和标记函项(labeling function)。支配是结点之间的一种关

系。结点 x 支配结点 y，仅当在树中存在一个从 x 到 y 的树枝的连接序列，记为⟨x, y⟩。支配关系是传递的、自返的和反对称的，简言之,是一种结点之间的弱偏序关系。若 x 与 y 是不同的结点，x 支配 y ,并且在 x 与 y 之间没有另外的结点，则 x 直接支配 y. 这时 x 被称为 y 的母亲结点， y 是 x 的女儿结点。被同一结点直接支配的若干结点称为彼此的姊妹节点。如(3)中被 S 直接支配的 NP 与 VP 就彼此称为姊妹结点。不被任何结点所支配的结点叫树根(root) ，如(3)中的 S 对应的结点。不支配任何结点的结点叫树叶(leaves) ，如(3)中的 my, sister 等对应的结点。其次，居前关系是结点之间从左到右的关系，这些结点之间不存在支配关系。如在(3)中，标记 Det 对应的结点在 NP 对应的结点之前。结点 x 在结点 y 之前也记为⟨x, y⟩ 居前关系是传递的、非自返的和非对称的，简言之，是一种结点之间的严格偏序关系。最后是标记函项的概念。由于每个结点对应唯一一个标记，这就形成了标记函项 L ，其定义域是树形图结点的集合，其值域是由语法范畴与词素作成的标记集合。

有了以上预备知识，便可从数学角度给出树形图的严格定义如下：

定义(树形图)一个成分结构树是这样一个数学结构⟨N, Q, D, P, L⟩ 满足：

N 是结点的有穷集合：

Q 是标记的有穷集合；

D 是 $N \times N$ 中的弱偏序关系，即支配关系；

P 是 $N \times N$ 中的严格偏序关系，即居前关系：

L 是从 N 到 Q 的函项，即标记函项。

并且使得下面的条件成立：

(i) $(\exists x \in N)(\forall y \in N)[\langle x, y\rangle \in D]$(单一根条件)

(ii) $(\forall x, y \in N)[[\langle x, y\rangle \in P \vee \langle y, x\rangle \in P] \leftrightarrow [\langle x, y\rangle$

$\notin D \& \langle y, x\rangle \notin D]]$（不相容条件）

(iii)$(\forall w, x, y, z \in N)[[\langle w, x\rangle \in P \& \langle w, y\rangle \in D \& \langle x, z\rangle \in D] \rightarrow \langle y, z\rangle \in P]$（非缠绕条件）

条件(i)说的是每个树形图存在着一个支配一切结点的结点，这就是树根。条件(ii)是说居前关系与支配关系不相容，两结点具有居前关系，就决不能具有支配关系，反之亦然。条件（iii)说明居前关系与支配关系也有某种微妙关系，即居前关系通过支配关系延续下去的关系。二者并非“纠缠不清”。如(3)中，V对应结点在NP对应结点之前，同时V结点又支配found结点，NP结点又间接支配unicorn结点，则有found结点在unicorn结点之前的关系。有了关于树形图的严格定义，也就产生了关于树形图的一系列定理，如：

定理1 给定树 $T = \langle N, Q, D, P, L\rangle$，每一对姊妹结点由P排序。

定理2 给定树 $T = \langle N, Q, D, P, L\rangle$，其树叶由P排成全序。

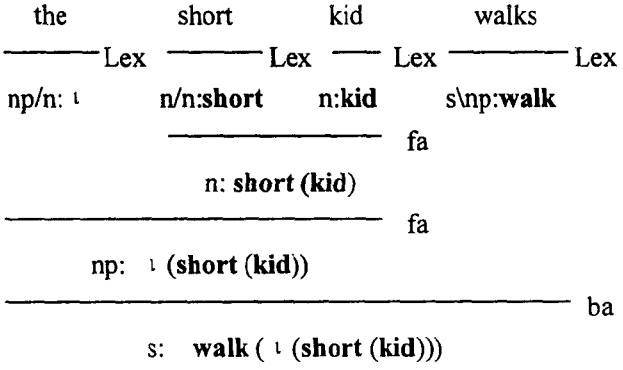
……

有关证明过程参见帕蒂[130] 第444页。

以上树形图方法在DRT等形式语义学研究流派中广泛运用。80年代以来，在范畴语法那里则流行另一种表现方式的树形图。这种树形图被称作是斯蒂德曼(Steedman)风格的图，其特征与上述有所不同。这种图树根在下而树叶在上，不同于上面所介绍的树根在上而树叶在下的树形图。斯蒂德曼树形图涉及以下预备概念：Tree指树的集合；BasExp指英语基本表达式，即英语词条的集合；Cat指英语句法范畴标记的集合，如{s, np, …}；Term

指与英语句法范畴标记对应的λ-词项的集合。[1]C 表示 Cat × Term 的子集，即由 Cat 中元素与相应的 Term 中元素作成的序对的集合。这里先给出一个斯蒂德曼树形图的例子：

(4)



(4) 的树根就是‘ s: **walk**(ι (**short** (**kid**)))” ；树叶分别是“ the”、“short”、“kid” 和‘ walks”。斯蒂德曼树形图的严格定义如下：

(i) BasExp ⊆ Tree ;

(ii) ⟨ C, ⟨ T_1 , …, T_n⟩⟩∈ Tree, 若 n ≥ o, C ⊆ Cat × Term , T_1 , …, T_n ∈ Tree 。

以上定义采用递归的方式，由英语基本表达式本身作为最小的树一步步生成较大的树，如由 the 是树生成 $\frac{\text{the}}{\text{np/n : ι}}$ 也是树。并且用序对的方式表示树，如上述树表示为：⟨np/n: ι , ⟨the⟩⟩。由于树

[1] λ-词项在范畴语法那里起表现英语表达式的逻辑结构的作用，详细介绍参见本编第四章 。

的定义的递归特征。以下刻画树的性质也都是递归定义，由 Tree 的起始定义 Tree 的派生，或者说，由最小的树具有某种性质定义较大的树具有某种性质。树 T 的结点集合 Node(T) 定义为：

(i) Node (e) = ∅ 若 e ∈ BasExp；

(ii) Node (⟨C, ⟨T_1, ⋯, T_n⟩⟩) = {C} ∪ Node (T_1) ∪ ⋯ ∪ Node (T_n), 若 T_1, ⋯, T_n ∈ Tree。

最初始的树 e 的结点集合为空集。在这里，树的结点相当于前面所说的与结点对应的标记。斯蒂德曼树形图没有标记的说法。树 T 的树叶集合 Leaf(T) 定义如下：

(i) Leaf (e) = {e}, 若 e ∈ BasExp；

(ii) Leaf (⟨C, ⟨T_1, ⋯, T_n⟩⟩) = Leaf (T_1) ∪ ⋯ ∪ Leaf (T_n), 若 T_1, ⋯, T_n ∈ Tree。

从上可以看出，树 T 的树叶集合就是树最上端的英语基本表达式的集合。就(4)而言，其树叶的集合 Leaf ((4)) = {the, short, kid, walks}。值得注意的是，树 T 的收获(yield), Yield (T) 的概念与树叶的概念很相似。其定义为：

(i) Yield (e) = e，若 e ∈ BasExp；

(ii) Yield(⟨C, ⟨T_1, ⋯, T_n⟩⟩) = Yield (T_1) ∪ ⋯ ∪ Yield (T_n), 若 T_1, ⋯, T_n ∈ Tree。

拿(4)来说，其收获 Yield ((4)) 为： the short kid walks。这里是英语词条的毗连序列，而不同于 Leaf ((4)) 所表现的英语词条的集合。树 T 的树根 Root (T)，其定义也是递归方式：

(i) Root (e) = e，若 e ∈ BasExp；