

doi:10.13582/j.cnki.1672-7835.2018.03.005

逻辑、语言和计算的交叉创新^①

邹崇理^{1,2}, 陈鹏³

(1.湘潭大学 哲学系,湖南 湘潭 411105;2.中国社会科学院 哲学所,北京 100732;
3.北京语言大学 信息科学学院,北京 100083)

摘要:逻辑学是自然科学和人文社会科学共同的基础学科,逻辑是构筑科学理论的工具。逻辑方法渗透到各门学科中导致创新驱动:逻辑和语言学的交叉研究产生了范畴语法的重大创新;逻辑与计算机科学的交叉融合引起了计算机科学的创新发展。在我国,逻辑针对语言学以及计算机科学的交叉研究尚明显不足,造成学科创新或理论创新的乏力,需要大力推进交叉创新研究。

关键词:逻辑学;交叉研究;创新

中图分类号:B81 **文献标志码:**A **文章编号:**1672-7835(2018)03-0033-09

1974年联合国教科文组织规定的七大基础学科依次为数学、逻辑学、天文学和天体物理学、地球科学和空间科学、物理学、化学、生命科学。确切讲,逻辑学是所有自然科学和人文社会科学共同的基础学科。任何科学理论都必须遵循逻辑学的基本原理,任何科学知识都可以采用逻辑方法进行分析。

逻辑学是一门可以渗透到任何学科中去的基础学科。一般来说,逻辑学的方法性质大都通过针对有关学科的交叉研究体现出来。在交叉研究中,逻辑学对有关学科的影响往往导致创新,形成新的思想或学科生长点。本文分别以逻辑和语言学的融合以及逻辑和计算机科学的互动为例来论述交叉创新的情况。

一 逻辑和语言学的交叉创新

在古希腊时代,逻辑学和语言学是同根同源的。当时的逻辑学和语言学研究均建立在古希腊文的基础上,逻辑的理论术语跟语法的理论术语很难区分。“古希腊时代逻辑上和语法上都把一个句子分为主词和谓词两部分,这是逻辑分析受

语言结构形式的影响,而语言的分析又是以逻辑分析为基础的最典型的例子”^①。作为西方的“逻辑之父”,亚里斯多德的著作《解释篇》致力于研究语言中的名词、动词和语句同逻辑命题之间的关系。而中世纪的逻辑学家认为,逻辑、语法和修辞学是具有紧密联系的课程“三艺”。到上个世纪,逻辑和语言学越走越近,交叉创新的趋势愈益明显,其例证主要有^②:

转换生成语法 + 内涵逻辑思想 = 蒙太格语法 MG;

自然语言量化现象 + 集合论思想 = 广义量词理论 GQT;

自然语言回指现象 + 动态逻辑思想 = 话语表述理论 DRT

语言学基于词汇的经验主义 + 逻辑基于规则的理性主义 = 组合范畴语法 CCG

下面我们就以组合范畴语法 CCG 的催生为例,来论述逻辑和语言的交叉创新。

语言学基于词汇的经验主义是一种从真实文

① 收稿日期:2018-02-19

基金项目:国家社会科学基金重大项目(17ZDA027)

作者简介:邹崇理(1953-),男,四川成都人,湘潭大学特聘教授,中国社会科学院哲学所研究员,博士生导师,中国逻辑学会会长,主要从事自然语言的逻辑研究。

①王维贤:《王维贤语言学论文集》,商务印书馆2007年版,第400页。

②生物学可以通过物种嫁接产生新物种,A物种 + B物种 = C物种,C兼有A和B的优良品质。我们用物种嫁接的方式来表述学科的交叉创新。

本的个案个例出发研究词条的方法。传统语言学和基于统计的计算语言学大都擅长这种方法。人类要使用语言,必须掌握构造语言的原子材料——单词或词条,这是学习一门语言首先要懂得的知识。一门语言的单词常用的有几千条,总数是几万乃至几十万条。语言学在浩如烟海的文献中搜集这些词条,根据词条出现的不同语境,确立它们的各种用法涵义,编纂各种各样的词典,这些都是词汇经验主义方法所做的工作。

语言学基于词汇的经验主义的特点是:依据真实文本的个案个例,个案个例的文本表达体现为各种不同的语境;词条是研究的中心,把自然语言丰富多彩的句法构造归结为具体词条的研究。即是说,具体词条的多种用法决定了多样的语句表达。如在宾州中文树库中收录的包含结构助词“的”的汉语句子有^①:

(1) 而台商对澳门的捐助,坦白说,还不如在珠海投资的港商。

(2) 虽然四十年的航天投资总额还不及前苏联、美国一年的航天投资。

(3) 春节前的一天下午,一群西装革履的台商来到东莞企石镇人民政府。

(4) 不同于上一代父母才开始接触ABC,如今英文已经是小学生生活的一部分。

(5) 这首歌的词曲作者是意大利著名作曲家巴廖克,在开幕式上巴廖克本人与60岁的男高音歌唱家雷斯科托共同演唱了这首歌。

这是包含结构助词“的”的5个真实文本句,“的”出现了9次,9次的语境是不同的,显示出9种不同的句法作用。如例句(5)中第一个“的”的语境为:上文是左边的量化短语“这首歌”,下文是右边的名词偏正词组“词曲作者”。“的”在这里的句法作用是:毗连左右构成更大的名词偏正词组“这首歌的词曲作者”。根据真实文本的具体语境确立词条的词例用法,“的”的9个词例用法决定了有关9个短语或句子的构造。词汇经验主义方法对于描述自然语言大规模的真实文本起到至关重要的作用,因为真实文本总是具体语境下的语言表达。

语言学单一的词汇经验主义是有缺陷的。自然语言中不仅有词条的表述,还有根据词条构成

的短语词组乃至句子。分析自然语言,不仅需要掌握词条的个案个例用法,还需要掌握词条生成短语词组和句子的规律。词条的用法是人们长期生活形成的固化产物,人们可以通过词典囊括进来。但是构造短语词组乃至句子的过程却是无穷多样的,尤其是短语和句子的意义是由词条的意义组合推演获得的,不存在囊括所有句子意义的“句典”。语言学基于词汇的经验主义方法不足以揭示短语词组和句子的生成过程,不足以揭示构造无穷多句子的普遍规律。

逻辑学基于规则的理性主义是一种崇尚普遍规律的研究方法。20世纪美国学者 Chomsky 指出,语言知识的本质在于人类的“心智-大脑”中存在着一套语言认知系统,这样的认知系统表现为数量有限的原则和规则体系,体现出人类语言的普遍性机制。逻辑学强调的理性主义规则观念更为精准,直接同推演和计算关联。数理逻辑思想的奠基者,德国哲学家和逻辑学家 Leibniz 认为“推理即运算”,逻辑的任务就是研究先验的普遍的思维运算规则。当今荷兰的逻辑和计算语言学家 Moortgat 主张,认知等于计算,语法就是逻辑,分析就是推演。而计算、逻辑和推演都跟理性主义的规则观念分不开,语言构造的普遍性机制通过规则体现出来。在逻辑学看来,刻画思维或语言普遍规律的规则是研究的中心。作为个案个例的真实文本和词条只是说明规则如何使用的辅助工具,结合具体语境的词例用法甚至干脆被抽象掉。

逻辑学认为,自然语言构造机制的本质特征是“依据有穷多规则去构造和理解无穷多的语句”。根据规则,人类能够构造或理解从来没有见到过或听到过的句子。自然语言中的复杂句子要有多长有多长,是无穷多的。对此逻辑确立了体现递归组合思想的规则,来构造生成自然语言的无穷多的复杂长句子,如:

(6) That men who were appointed didn't bother the liberals wasn't remarked upon by the press.

(7) That everything you learned about America's history is wrong is known to the public.

(8) The man such that he loves a woman such that she hates a boy chants.

^①参见:CCGBank 语料库 www.ccgbank.net。

CCG 中表现为:衔接宾州中文树库对词条个案个例用法的研究结果。挑战“一词一范畴”的传统做法,根据不同的语境,给同一个词条指派不同的

范畴。如给前例句(1)((5)中结构助词“的”的9次出现,分别指派了9个不同的范畴以显示9个不同语境下的个案个例用法:

例句	第一个“的”的范畴	第二个“的”的范畴
(1)	PP\ \backslash (NP/NP)	(NP\ \backslash S)\ \backslash (NP/NP)
(2)	QP\ \backslash ((NP/NP)/(NP/NP))	QP\ \backslash (NP/NP)
(3)	LCP\ \backslash ((S/S)/(S/S))/(S/S)/(S/S))	(NP\ \backslash S)\ \backslash (NP/NP)
(4)	NP\ \backslash (QP/QP)	-
(5)	NP\ \backslash (NP/NP)	QP\ \backslash NP

自然语言的词条往往具有多种语境下的多种用法,一个词条对应多个词例,不同的词例用不同的范畴表示,“一词多范畴”的指派方法应运而生。社科中文 CCG 树库表明,词条对应范畴数量最多的前 10 名词条如表 1。

表 1 社科中文 CCG 树库的词条词例统计

词条	范畴数量(词例)	在树库中出现的次数
的	181	38 354
在	97	9 622
是	79	7 680
一	76	6 086
到	69	1 842
有	61	3 784
(57	935
上	53	2 232
为	52	2 366
了	51	6 164

汉语真实文本中的常用词使用频率非常高,其用法数量也是惊人的。结构助词“的”,使用频率多达 38 354 次,在 181 个语境下有 181 个具体用法,按照范畴逻辑的传统语词分类标准,一类基本语词对应一个范畴,“的”就属于 181 个基本语

词类。社科中文 CCG 树库的词条总共是 46 085 条,当然词条所属范畴有不少是交错共享的。但可以想见,树库的基本语词类不是小数目,清华中文 CCG 库的基本语词是 763 类,蒙太格语法的基本语词只有 9 类,这就是 CCG 所谓的大词库,也是自然语言处理领域内的“大数据”思想。

逻辑基于规则的理性主义优势在社科中文 CCG 库中也有突出的表现:其核心是关于函子范畴的一系列组合推演规则,这些规则描述了自然语言构造的普遍规律,这些规则出自范畴逻辑的结构公设及其定理,这些规则是逻辑主张的函项运算思想的传承,是逻辑理性主义精神的彰显。以下是社科中文 CCG 库中常用的组合推演规则:

需要指出的是:社科中文 CCG 树库使用的逻辑规则具有跟其他自然语言共享的普遍性。不同的自然语言其差异主要体现在不同的词库设置上,逻辑规则的使用大致相同,且常用规则的数量也不多。这就是 CCG 所谓的“小规则”设计。然而规则数量虽少,但规则的推出能力却非常大,可以推出由词条构成的词组短语,乃至无穷多的语句,甚至可以推出短语和句子无穷多的意义。这就是 CCG 显示出的逻辑基于规则的理性主义优势^①。

函子范畴的右贴合

$$X/Y \quad Y \Rightarrow X$$

函子范畴的向前组合

$$X/Y \quad Y/Z \Rightarrow X/Z$$

函子范畴的向前交叉组合

$$X/Y \quad Y\backslash Z \Rightarrow X\backslash Z$$

函子范畴的向前置换

$$(X/Y)/Z \quad Y/Z \Rightarrow X/Z$$

函子范畴的左贴合

$$Y \quad X\backslash Y \Rightarrow X$$

函子范畴的向后组合

$$Y\backslash Z \quad X\backslash Y \Rightarrow X\backslash Z$$

函子范畴的向后交叉组合

$$Y/Z \quad X\backslash Y \Rightarrow X\backslash Z$$

函子范畴的向后置换

$$Y\backslash Z \quad (X\backslash Y)\backslash Z \Rightarrow X\backslash Z$$

例如,在社科中文 CCG 库中,对前文例句(1)

的句法分析树大量采用了“函子范畴的左贴合”

^①在 CCG 树库基础上开发的句法分析器可以分析推演无穷多的语句,宾州英文 CCG 树库已经做到了这点。清华中文 CCG 库和社科中文 CCG 库的句法分析器有待开发,目前仅限于分析推演有穷数量的句子。

和“函子范畴的右贴合”规则(见图 2)。

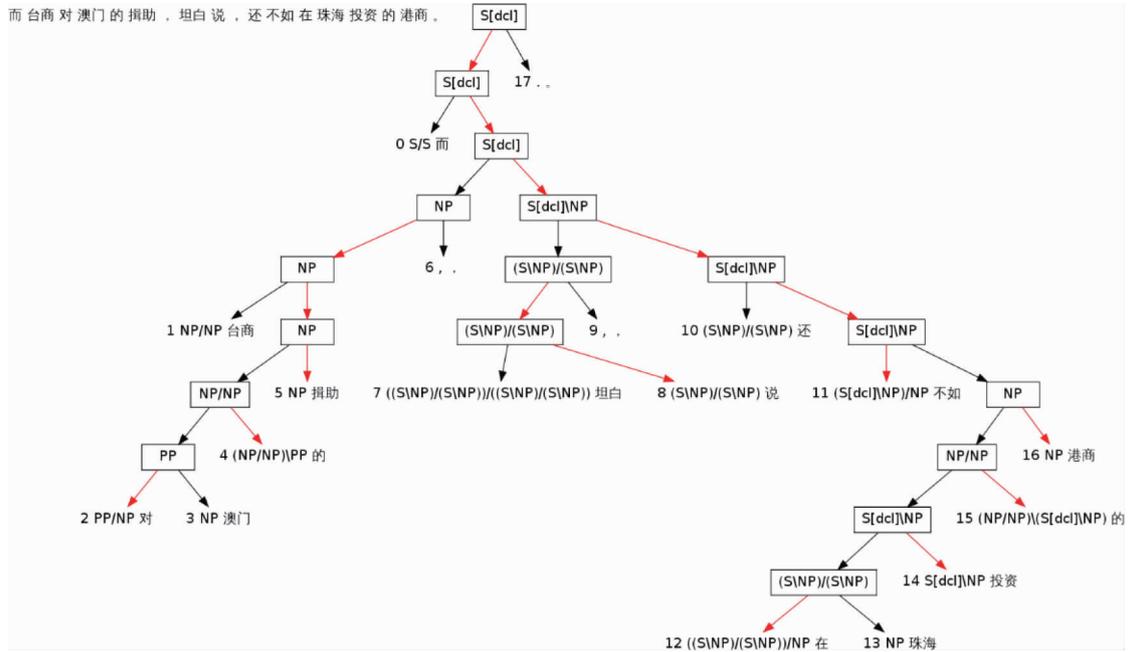


图 2 例句(1)的句法分析树

由于 CCG 兼具语言学基于词汇的经验主义和逻辑学基于规则的理性主义两方面的优势,其创新价值不言而喻。CCG 的研究较大幅度满足了计算机处理大规模真实文本的需要,根据 CCG 设计的计算机自然语言句法分析器在诸多形式语言学理论的自动分析中是速度最快的。“在 2009 年约翰霍普金斯大学举行的夏季研讨班上,研究人员通过采用优化的句法分析算法,使 CCG 句法分析在维基百科语料上达到每秒超过 100 句的分析速度”^①,而基于中心语驱动语法的计算机处理软件几秒钟才能完成一个语句的分析。

二 逻辑和计算机科学的交叉创新

追溯现代计算机科学的起源,与逻辑有着密不可分的关系。众所周知,自从罗素与怀特海共同撰写《数学原理》之后,对数理逻辑的研究蓬勃兴起,人们甚至期望以逻辑为基础,构建整个数学,乃至科学大厦。在这种逻辑主义的驱使下,不可避免地需要对“能行可计算(effectively calculable)”概念进行形式化。在“能行可计算”概念的探索中,逻辑学家丘奇和哥德尔,计算机奠基者图

灵几乎在同时间给出完全不同且又相互等价的定义。丘奇发明了 Lambda 演算,用来刻画“能行可计算”。哥德尔提出“一般递归函数”作为对“能行可计算”的定义。图灵则通过对一种装置的描述,定义“能行可计算”的概念,这种装置被后人称作“图灵机”,这正是现代计算机的理论模型,标识现代计算机科学的诞生。

逻辑在计算机科学的创新发展中,扮演了非常关键的作用。逻辑甚至被称作为“计算机科学的演算”^②。逻辑和计算机科学的交叉创新的例证有:

布尔逻辑 + 开关电路 = 计算机集成电路理论;

逻辑证明论模型论 + 计算机软件系统 = 计算机系统正确性验证理论;

现代逻辑 + 编程语言 = 程序设计语言理论。

我国著名逻辑学家莫绍揆先生指出:“事实上,程序设计或者就是数理逻辑,或者是用计算机语言书写的数理逻辑,或者是数理逻辑在计算机上的应用。”^③以下以现代逻辑和计算机编程语言

①宋彦,黄昌宁等:《中文 CCG 树库的构建》,《中国计算语言学前沿进展》2011 年第 3 期。

②Manna Z, Waldinger R. *The Logical Basis for Computer Programming*, Boston: Addison-Wesley, 1985, pp.67-72.

③王元元:《计算机科学中的现代逻辑学》,科学出版社 2001 年版,第 i-6 页。

的融合促使程序设计语言理论的发展为例,来说明逻辑和计算机科学的交叉创新。

现代逻辑中对计算机编程语言产生较大影响的分支是一阶逻辑、逻辑类型论和逻辑语义学。一阶逻辑是德国逻辑学家弗雷格在《概念文字》一书中提出的。弗雷格认为一阶逻辑是一种“概念文字”,是“纯粹思维的形式语言”。弗雷格首次提出形式语法的概念,将逻辑演绎归结为形式推导。正是弗雷格对逻辑的这种“语言化”处理,开创了对形式语言构造和处理的新方式。逻辑类型论是英国逻辑学家罗素为解决悖论问题提出的一种理论,这种理论对高阶逻辑中各种层次的逻辑表达式进行类型区分,对各类表达式的组合进行限制。逻辑语义学是逻辑学家塔斯基为逻辑系统确立的一种形式语义理论。给逻辑表达式提供指称语义,特别地,对逻辑表达式中的公式给予的是真值条件语义。

计算机编程语言是计算机科学的核心研究对象,其起源来自图灵那篇具有革命性意义的论文《论可计算数及其在判定问题中的应用》中提出的“图灵机”概念。在图灵机的构成中,主要包括“自动机”和“指令表语言”,其中指令表语言是指描述状态转换表的语言,即,描述自动机状态转换、读写以及移动的语言。这里的指令表语言,就是我们现在编程语言的雏形。最早指令表语言的语法描述是以有限状态自动机的方式,例如:我们可以将指令表语言的语法描述为一个五元组:

$$\langle Cstat, Csym, Nsym, Dir, Nstat \rangle$$

其中,Cstat表示当前的状态,Csym表示当前的符号,Nsym表示新的符号,Dir表示移动的方向,Nstat表示性的状态。此外我们可以定义状态集合、符号集合、初始状态、终止状态,等等。

相比逻辑学的研究,计算机科学是一门非常年轻的学科,从1947年第一台计算机诞生之日算起,到现在为止,也不过只有70年的发展历史,比较具有上千年的逻辑学发展而言,计算机科学的发展还处于初级阶段。应该说,吸取丰富的逻辑学营养,结合计算机学科的一些新特性,就能够推动计算机学科的创新。逻辑为什么能够对计算机编程语言产生作用?这里交叉创新的切入点是:逻辑语言和编程语言都是人工语言,其共同点是构造机制都是人为规定的,所以逻辑语言的精

准表述方式自然就可以移植到程序设计语言。作为一种人工符号语言,程序设计语言的发展创新就与逻辑密不可分。因此,当今上千种编程语言无一例外都借鉴和沿用了一阶逻辑对形式语言的构造和处理方法。在通常的程序语言设计中,需要明确对表达式类型的指派。而逻辑语义学发展出的结构操作语义能够较好地描述计算机程序语言的状态转换机制。可以说,程序设计语言理论是将一阶逻辑、逻辑类型论和逻辑语义学等现代逻辑理论与编程语言交叉融合导致的创新产物。逻辑对编程语言的作用表现为:编程语言的形式语法(formal syntax)、编程语言的类型系统(type system)和编程语言的形式语义学(formal semantics)。

编程语言的形式语法是指用逻辑的方式描述的编程语言语法。逻辑的形式语法思想源于弗雷格的一阶逻辑。形式语法的方法在埃米尔·珀斯特(Emile Post)那里得到发展,珀斯特在研究机械式算法过程中,采用产生式方法来描述一般的字符串操作,珀斯特产生式(Post's Production)成为形式语法方法的基础。约翰·巴克斯(John Backus)在珀斯特产生式的基础上,用巴克斯范式(Backus-Naur Form)开发并描述了编程语言ALGOL 58。minML是哈尔彭(Joseph Y. Halpern)等人开发的编程语言^①,是对ML语言裁剪的一个展示片段。minML的巴克斯范式描述如下:

$$e ::= x \mid n \mid e_1 \circ e_2 \mid \text{true} \mid \text{false} \mid x = e_1 \mid e_1 == e_2 \mid \text{if } e \text{ then } e_1 \text{ else } e_2 \mid \text{fun } f(x:\tau_1):\tau_2 \text{ is } e \mid e_1(e_2)$$

不难看出,相比于之前的指令表语言的语法描述方式,通过逻辑的形式语法方式描述编程语言更准确且更容易理解,在minML的形式语法描述中,我们清楚地见到了编程语言表达式的定义与构造、函数的构造、分支选择等语法特性。这种简约精准的表述方式是对程序设计语言理论的创新。

在20世纪80年代和90年代之间,随着逻辑和计算机科学思想的融会,促进了编程语言的类型理论的发展。编程语言的类型系统是对程序的构成实施某种约束,从而保障程序不会因值的不当解释而发生错误。雷诺兹(John C. Reynolds)将编程语言的类型系统定义为“强制执行抽象层

^①Halpern J Y. "On the Unusual Effectiveness of Logic in Computer Science", *Bulletin of Symbolic Logic*, 2001(2):213-236.

次的句法”^①。借鉴逻辑类型论的风格,编程语言 minML 的类型系统表述如图 3。

$\overline{\Gamma \vdash x: \Gamma(x)}$	
$\frac{\Gamma \vdash x: \text{int} \quad \Gamma \vdash e_1: \text{int}}{\Gamma \vdash x=e_1: \text{int}}$	$\frac{\Gamma \vdash x: \text{bool} \quad \Gamma \vdash e_1: \text{bool}}{\Gamma \vdash x=e_1: \text{bool}}$
$\overline{\Gamma \vdash n: \text{int}}$	$\frac{\Gamma \vdash e_1: \text{int} \quad \Gamma \vdash e_2: \text{int}}{\Gamma \vdash e_1 \circ e_2: \text{int}}$
$\overline{\Gamma \vdash \text{true}: \text{bool}}$	$\overline{\Gamma \vdash \text{false}: \text{bool}}$
$\frac{\Gamma \vdash e_1: \text{int} \quad \Gamma \vdash e_2: \text{int}}{\Gamma \vdash e_1 = e_2: \text{bool}}$	$\frac{\Gamma \vdash e: \text{bool} \quad \Gamma \vdash e_1: \tau \quad \Gamma \vdash e_2: \tau}{\Gamma \vdash \text{if } e \text{ then } e_1 \text{ else } e_2: \tau}$
$\frac{\Gamma, f: \tau_1 \rightarrow \tau_2, x: \tau_1 \vdash e: \tau_2}{\Gamma \vdash \text{fun } f(x: \tau_1): \tau_2 \text{ is } e: \tau_1 \rightarrow \tau_2}$	$\frac{\Gamma \vdash e_1: \tau_1 \rightarrow \tau_2 \quad \Gamma \vdash e_2: \tau_1}{\Gamma \vdash e_1(e_2): \tau_2}$

图 3 minML 的类型系统

可以看出,在编程语言 minML 的类型系统中,函数类型的构造规则与逻辑的根岑系统的蕴涵引入与消去规则极其相似。这种相似性并非偶然,实质上,编程语言的类型系统体现了逻辑类型论的精神,潜在地具有柯里-霍华德同构(Curry-Howard isomorphism)所揭示的思想:计算机程序与逻辑证明之间存在紧密的关联。编程语言的类型理论表达了明晰的数据抽象、多态和继承等高级编程语言的特性,从而为编程语言的设计、分析和实现提供了新的概念框架。

我国著名计算机科学家周巢尘说过:“形式语义对编程语言的设计、使用和实现产生了深刻的影响,语言的形式语法和形式语义(统称语言的形式定义)已经成为编程语言的必要组成部分。”^②编程语言的逻辑形式语义主要包括指称语义、操作系统和公理语义等。采用普洛特金(Plotkin)的结构化操作语义,我们可以给出程序如何执行的语义,将操作语义定义为抽象机器的状态之间的转换,编程语言的结构化操作语义完全遵循逻辑的演算方式,以逻辑证明的方式给出程序状态空间转换过程的描述,是逻辑语义学针对编程语言核心概念的创新。详细介绍参见《结构化操作语义的起源》^③,这里从略。

可以说,现代计算机科学的发展,应用是牵引,而逻辑则是引擎。正如马丁·戴维斯在《逻辑

的引擎》一书的前言部分所言:“20 世纪 50 年代,计算机还是塞满整个房间的庞然大物,而时至今日,它已经演变成轻巧而强大的机器,能够完成各种意想不到的任务。虽然经历了巨大的演变过程,计算机所基于的逻辑始终如一。这些逻辑概念在几个世纪以来一群天才思想家一步步的努力下发展而来的。”^④逻辑和计算机科学的交叉研究是计算机科学创新的引擎,是计算机科学发展的原动力。

三 我国逻辑、语言和计算的交叉研究不足导致创新乏力

近十多年来,我国学术界在逻辑、语言和计算的领域,虽然出现了一些打破学科界限的交叉研究苗头,但总体上看,这些交叉研究和国外比起来差距很大。逻辑牵头的交叉研究在当今我国因为逻辑意识的淡泊和根深蒂固的学科分界而受到漠视,从而导致理论和技术创新的匮乏。

爱因斯坦说:欧洲科学技术的迅猛发展得力于两门学科,一门是亚里士多德创立的形式逻辑,一门是培根创立的实验科学。1977 年《大英百科全书》把逻辑学列为知识的五大分科之首,即:逻辑学、数学、科学(自然科学、社会科学和技术科学)、历史学和人文学(语言文学)、哲学。而我国在学科的课程设置和科研的投入上却严重漠视和忽略逻辑,逻辑学理论上的地位和实际情况很不相称。按照联合国教科文组织的规定,逻辑学位于数理化天地生基础学科系列中的第二位,其他基础学科的内容大都在我国小学中学时就开始涉猎,而逻辑学在中小学教育中却几乎没有踪影。大学中的情况也不容乐观,哲学系大都开设了逻辑课,而在与逻辑密切相关的数学、语言和计算领域,基本上没有逻辑课程的设置。在国家社科基金的学科分类中,逻辑仅仅是哲学下属的 2 级学科,哪有同数理化比肩而立的伟岸地位?我国的逻辑学科根底不深,教育基础太浅,我国大学普遍“重理工轻人文”,哲学就不起眼,在哲学麾下的逻辑就更卑微了,很难想象如此卑微的学科能够牵头搭建跨学科的研究平台,以逻辑学为导引的

^①Reynolds J C. “Three approaches to type structure”, *Proc. of the International Joint Conference on Theory and Practice of Software Development*, New York: Springer-Verlag, 1985, pp.97-138.

^②周巢尘:《形式语义学引论》,湖南科学技术出版社 1985 年版,第 i 页。

^③Plotkin G D. “The origins of structural operational semantics”, *Journal of Logic & Algebraic Programming*, 2004 (July - December): 3-15.

^④马丁·戴维斯:《逻辑的引擎》,张卜天译,湖南科学技术出版社 2013 年版,第 i 页。

交叉研究势必举步维艰。

反观国际上,逻辑、语言和计算的交叉研究开展得如火如荼。如,荷兰阿姆斯特丹大学的“逻辑、语言和计算研究所”设置了“逻辑和语言”“逻辑和计算”和“语言和计算”三个跨学科的研究方向,多年来培养了200多名博士和一大批硕士,在世界各地从事专业研究工作和教学工作。而国内高校科研院所的学科设置大都界限分明壁垒森严,各自单打独斗,真正意义上的交叉研究不多。我国学者的知识大都比较单一化,逻辑学者一般不懂语言学和计算机科学,计算机科学的较多专家也没有多少现代数理逻辑的知识,语言学者中能够通晓逻辑的就更少了,国内大多数语言学专家对逻辑倡导的形式化方法不是积极认可的,逻辑、语言和计算的交叉研究在我国的开展是比较乏力的。

20世纪下半叶以来,在欧美发达国家,逻辑和语言学的交叉研究呈现生机勃勃的局面,出现了一系列原创性的理论。如语言逻辑领域的蒙太格语法 MG、广义量词理论 GQT、话语表述理论 DRT、范畴类型逻辑 CTL 和组合范畴语法 CCG 等。这些理论的原创均是欧美学者所为,而我国的学者少有实质上的跨学科交叉研究,研究人员稀少,研究经费短缺,理论上的原创几乎没有。

在计算机科学领域,即使人员和经费不成问题,然而我国由于缺乏逻辑学和计算机科学的交叉研究,也极大影响了计算机科学的创新。毫无疑问,计算机科学是当今世界发展最快,对世界影响最为深远的一门学科。计算机科学理论创新的主流在国外,不少创新成果的获得跟逻辑的交叉研究分不开。图灵奖自1966年设立以来,已经成为计算机科学领域的最高荣誉,被称为“计算机界的诺贝尔奖”。在1996年至2015年期间60项图灵奖获奖成果中,有接近20项的成果与逻辑有直接的关联,是逻辑学与计算机科学交叉创新的所绽放的绚烂花朵^①。例如:阿米尔·伯努利(Amir Pnueli)把时态逻辑方法引入计算机科学的研究成果获1996年计算机图灵奖。埃德蒙·梅尔森·克拉克(Edmund Melson Clarke)与艾

伦·爱默生(E. Allen Emerson)和约瑟夫·斯发基斯(Joseph Sifakis)共同基于逻辑中的模型论开发了针对计算机软件和硬件验证的模型检测技术,并藉此获得2007年度的图灵奖,等等。而我国的计算机科学界尚没有实现图灵奖零的突破,缺乏同逻辑的交叉研究是原因之一。

计算机及信息产业成为全球技术创新的主要动力。我国的计算机及其信息产业已经成为国家的支柱型产业^②。然而在我国计算机及信息产业领域却存在着大而不强,缺乏核心竞争力的尴尬局面。要提升其核心竞争力,无疑要先提升我国在计算机科学与技术领域的创新能力,而计算机科技创新能力的提升又跟逻辑精神的熏陶分不开。仅就编程语言而言,全球目前主流的编程语言有上千种,几乎每一个欧美国家,甚至日本,都发明过通用的编程语言^③。而我们国家有着全球数量居第二的程序员数量^④,但没有一种编程语言是由中国大陆的专业人员发明的。原因之一是,我国的计算机专业教育几乎不涉及数理逻辑的教育,数理逻辑在国内几乎被排挤到必修课程之外。而著名的计算机软件设计大师戴克斯特拉(Dijkstra)曾说:“假如我早年在数理逻辑上好好下点功夫的话,我在搞软件时就不会犯这么多错误。不少东西逻辑学家早就说了,可我不知道。要是我能够年轻20岁,就要回去学逻辑。”^⑤反观我国,计算机科学的高等教育对逻辑采取严重的忽略和漠视态度,这就极大限制了计算机科学和逻辑学相融合的交叉研究,也就大大削弱了国内计算机专业人员对于编程语言的创新能力。

四 关于交叉创新的进一步工作

综上所述,逻辑跟语言学及计算机科学的交叉研究对于学科创新和技术创新具有毋庸置疑的重大意义,而我国的总体情况不容乐观。但是危机孕育机遇,困境蕴涵新生。在本文结束时我们提出两点思考。

前文论述到:语言学基于词汇的经验主义 + 逻辑学基于规则的理性主义 = 组合范畴语法 CCG,包括中国社科院哲学所在内的国内外学者

①历年图灵奖的获得者及其获奖成就的介绍可以参考美国计算机协会的图灵奖网站:<http://amturing.acm.org>。

②据国家工业与信息部的统计数据,2015年,信息产业占国家GDP比重为26%。

③究竟出现过多少计算编程语言?这并没有一个统一和精确的说法, Bill Kinnersley 提出一个比较全的编程语言列表,收录了大概2500种编程语言,参见 <http://www.info.univ-angers.fr/~gh/hilapr/langlist/langlist.htm>。

④根据国际数据公司(IDC)在2015年的统计,中国大约有200万名程序员,居全球第二,排名第一位的是美国。

⑤王元元:《计算机科学中的现代逻辑学》,科学出版社2001年版,第iii页。

在此已经获得了创新成果。但是,直到目前为止,大多数学者的 CCG 研究对自然语言语义的关注还不够充分,2009 年在约翰·霍普金斯大学用 CCG 对维基百科语料的测试也只是句法分析。自然语言的语义相比句法更为复杂,构建句法和语义并行推演的 CCG 树库比构建纯粹的句法分析 CCG 树库困难得多。在国际上,CCG 的句法分析器问世多年后,2015 年才由 Mike Lewis、Luheng He 和 Luke Zettlemoyer 等学者提出世界上第一个关于英语的句法语义分析器及 CCG 句法和语义推演的自然语言处理系统 EasySRL^①。追踪最新前沿刻不容缓,我们打算顺势而为确立新的交叉研究:社科中文 CCG 句法树库 + 逻辑语义学思想(依据蒙太格语义学和事件语义学等提供的分析技术)= 中文 CCG 句法语义树库,进而开发中文句法语义分析软件。遵循蒙太格语义学和事件语义学等给句法规则例对应语义规则的做法,对社科中文 CCG 树库的上千条规则例,逐条匹配语义规则,当然有许多语义疑难情况需要特殊处理,困

难多,工作量大。我们应该有信心和耐心去履行这样的使命。

前文提到的创新成果还有:现代逻辑 + 编程语言 = 程序设计语言理论。从交叉创新的视角看,我国学者唐稚松院士^②所创立的 XYZ/E 编程语言是在时序逻辑系统中引入状态转换控制机制的产物,是时序逻辑交叉融合可执行命令式编程语言的成功案例。从唐院士提出 XYZ/E 后的 20 多年期间,计算机软工技术已经发生了巨大的变化,编程语言已由当年的不足百种发展到上千种,相对而言,XYZ/E 已经有些陈旧,不完全能够适应现代软工发展的需要。同时,时序逻辑和动态逻辑在此期间也获得长足发展,如何在时序逻辑的框架中采用动态逻辑的方式更精准地揭示状态转换的动态思想,有待今后的工作。面对计算机科学高速发展的机遇,我们不妨进行新的尝试,沿着“逻辑与计算机科学的交叉创新”的路子走下去,期待取得进一步的创新成果。

Interdisciplinary Study Between Logic, Linguistics and Computing: An Innovative Perspective

ZOU Cong-li^{1,2} & CHEN Peng³

(1. Department of Philosophy, Xiangtan University, Xiangtan 411105, China;

2. Institute of Philosophy, Chinese Academy of Social Sciences, Beijing 100732, China;

3. School of Information Science, Beijing Language and Culture University, Beijing 100083, China)

Abstract: As a fundamental tool for theorizing efforts, logic is a basic discipline for both natural sciences and humanities and social sciences. The application of logical methods in various branches of academic pursuit has produced quite a number of significant innovations, for example, the fusion of logic with linguistics brings about the rebirth of categorial grammar, and the fusion of logic with computer science gives rise to great advances in the latter field. In China, the explorations made by researchers in the above two areas are regrettably inadequate, which needs to encourage innovative work in the concerned research areas.

Key words: logic; fusion of disciplines; innovation

(责任校对 龙四清)

①参见: <https://github.com/uwnlp/EasySRL>。

②唐稚松(1925-2007)大学与研究生就读于清华大学哲学系,师从我国著名逻辑学家金岳霖先生学习逻辑学,后进入中国科学院计算技术研究所工作,从事逻辑学与计算机科学的交叉研究,于1991年当选中国科学院院士。唐稚松提出的XYZ/E创新性成果荣获1989年国家自然科学奖一等奖和1996年何梁何利科学技术进步奖,是我国软工领域发展的一个里程碑。