程序设计语言

——语法与语义

薛少庆 等著

四川教育出版社

程序设计语言——
语法与语义

四川教育出版社
一九九〇年·成都

责任编辑：赵鑫辉
封面设计：杨　林
版面设计：刘　江

程序设计语言——语法与语义

四川教育出版社出版发行　（成都盐道街三号）
四川省新华书店经销　攀枝花新华印刷厂印刷

开本787×1092毫米 1/32　印张7.5　插页 4　字数160千
1990年1月　第一版　1990年1月　第一次印刷
印数　1——625　册

ISBN7-5408-1129-3/G·1100　定价：2.67 元

●前　言

程序设计语言是计算机科学的理论和实践中经常使用的一种重要工具。每一个计算机工作者，无论是搞理论的还是搞应用的，搞硬件的还是搞软件的，无论是计算机系统的设计制造者还是用户，无例外地都免不了要和它打交道。在浩如烟海的计算机科学教程中，讲授各种语言的课本恐怕是为数最多的。学会一种特定的程序设计语言，能使用它达到自己的某种目的，这固然重要；但掌握程序设计语言的共同特点，在比较中学习它们，把它们作为一个整体对象研究、学习，也是很有好处的。程序设计语言学，计算机科学这一枝叶繁茂的大树上萌发成长的茁壮分枝，就是把程序设计语言作为整体来研究其共性的新兴学科。这一学科由程序设计语言的语法学、语义学、语用学构成。它侧重基础理论研究，而对程序设计的实践又有极大的指导意义。可以说，掌握程序设计语言学的基本知识是计算机专业工作者，特别是软件工作者的基本素养。目前，国内出版物中多的是各种语言的教本，而讲授程序设计语言学的书却属凤毛麟角。本书的目的就是简单介绍程序设计语言学的基本知识以作为学习这方面

理论的导引。作者的宗旨是：力求讲清基本问题，对艰深的课题只作浅近介绍而不穷究其底蕴。目的是：一方面使具有较低理论背景的人读了这本小书后也会有所收获，另一方面又能为希望在这一领域进一步攀登的人提供一方垫脚石。

由于程序设计语言语用学与语言的实现关系密切，而基于其重要性，语言的实现又是必须专门论述的问题，所以我们基本上不讨论语用（只是必要时偶尔提及），而把语法和语义作为主要讨论对象；又因为论述语法方面的出版物很多（任何一本讲编译方法的书都用很大篇幅来讨论语法问题），因此，我们把侧重点放在语义问题的阐述上。

我们的内容安排是这样的：

第一章为引论，介绍语言、语法、语义、程序设计语言等基本概念，讨论程序设计语言设计和定义的一些原则。

第二章以叙述形式语言学的结果为主研究程序设计语言学的语法学问题，重点是程序设计语言语法学中的上下文无关文法及其子类。

第三章讨论程序设计语言的语义学问题，介绍操作语义、标志语义、公理语义、代数语义等几种主要语义模型。

第四章通过对一种著名的元语言META-Ⅳ的介绍来较详细地论述语义学的标志方法。

本书由赖少庆、计科编写，陆汝铃审阅。本书的基础系赖少庆在厦门大学讲学时所用讲义。陆汝铃、魏道政、周巢尘、C.B.Jones和张少润、蔡经球等教授曾给作者以极大的帮助和支持，在此谨向他们表示谢意和致意。

作者

●目　录

●第一章 引 论

程序设计语言与其它任何语言一样，也是由语法和语义两大基本成分构成的。为了精确地定义程序设计语言并有效地实现它，应用它，就必须对程序设计语言的各构成要素作深入的研究，以便找到严格定义语法和语义的好方法（也就是设计程序设计语言的好方法）。这本小册子的第二、三两章就讨论这方面的内容。作为这些讨论的准备工作，应当对语言、语法、语义、语用这些基本概念，对程序设计语言的概念，对判断什么是好的程序设计语言的准则，有一个基本的了解。引论的目的就是要给出有关这些概念的知识。

第一节 语言、语法与语义

当今世界里，电子计算机已经渗透到各个领域。成了现代科学技术、现代化的大工业生产，乃至现代化的生活所不可缺少的助手。现在，人们已经不再把电子计算机单纯地当作一种计算工具，而是把它看成一种几乎无所不能的信息处理工具，是可以在许多场合下代替人脑工作的电脑。计算机

究竟能否达到人的智能高度，这是颇多争议且难获解答的问题，但从信息处理的角度来看，计算机的出现却是实质性的一大进步，由此可以发展出许多新的观点。人们要利用计算机处理信息，首先就得以一定的方式表示信息，并把这些信息传送给计算机。一般地讲，这些信息表现为某种形式的语言。事实上，一个计算机系统就是一个能解释语言的机器。计算机接受语言的句子，理解它，服从它，根据它的要求完成一些动作，最后给出结果回答问题。可以毫不夸张地说，计算机系统完成其任务的整个过程都是建筑在以某种计算机语言所表达的程序（我们把为计算机规定的执行动作序列称为程序）的基础之上。这种把计算机看作是能解释语言的机器的观点决定了计算机语言在计算机世界里的重要地位，使计算机语言，特别是程序设计语言，成为计算机科学中人们兴趣最浓、研究最多、结果最富的领域之一。

作为一种语言，尽管是人造的语言，程序设计语言也服从语言学的一般规律，实际上程序设计语言与自然语言（英语、汉语等）也确有许多共性。由于这种缘故，在讨论程序设计语言之前，有必要弄清楚语言等几个更一般的概念。

语言

什么是语言？按照语言学上具有极大权威的 Webster 的定义，语言是“为相当大的团体的人所懂得并使用的字以及组合这些字的方法的统一体”。这个定义基本上抓住了问题的要害：语言的意义（人所懂得的字）和形式（组合这些字的方法）的统一体，也就是语法和语义的统一体。

Webster 对语言的定义适宜于任何种类的语言（自然语言和人为语言），当然也适合程序设计语言（它只是人为语

言的一种)。但是，程序设计语言比自然语言简单得多，形式化得多，有关它的知识必须更精确(或者说更刻板，否则就不能为死机器所理解)，所以，对于研究程序设计语言来说，这个定义显得太笼统。

为了更科学更严谨地研究语言的理论，人们发展了符号语言学(Semiotics，或译形式语言学)，试图为语言(包括自然语言)的研究建立更形式的基础。在符号语言学中，为语言的概念作了远较形式而严格的规定。

为给语言下一精确定义，首先要有字母表或符号集的概念，这就是：符号的任何有穷集。虽然有不可数的无穷多个符号存在，但我们只考虑可数无穷子集，并从这样的子集中抽出我们要求的有穷集来。字母表可以包括数字，大写和小写的拉丁和希腊字母，$、£、#等特殊等号，以及使用者希望有的任何符号。例如，

{0，1}

{A，B，…，Z}

{α，β，…，ω}

{人，饭，吃}

等都是合法的字母表。

然后，是句子的概念。一个字母表上的句子是由字母表的符号组成的有限长度的任何行。句子的同义词是行、串、字。句子称为语言的一个语法单位(有时又称为语法范畴)，还可以有其他的语法单位，以此来表示符号构成的不同复杂程度的级别。例如，我们经常使用的分句、复句、短语、程序等都是这样的一些语法单位。任何字母表上均可以有的一个特殊句子是空句子，即不含任何符号的句子，记为ε。设

V是一个字母表，我们用V^*表示由V的符号所组成的所有句子的集合，包括空句子ε。另外，用V^+表示集合$V^*-\{\varepsilon\}$。因此，若$V=\{0, 1\}$，则

$V^*=\{\varepsilon, 0, 1, 00, 01, 000, \cdots\}$

$V^+=\{0, 1, 00, 01, 000, \cdots\}$ （ε不在V^+中）

在以上定义的基础上，我们可给语言下这样一个定义：

语言是一字母表上的句子的任何集合。

也就是说，对于字母表V，V^*的任何一个子集均构成一个语言。比如说，上述全部二进制数的集合$V^+=\{0,1,\cdots\}$就构成一个语言，我们称它为二进制数语言。

应当承认，就揭示问题的实质而言，这一定义并不比Webster的定义高明。第一，它全然没有接触到语言的语义。第二，有关语法，它也并没有告诉我们多少东西：句子的任何集合，这种限定确实也太空泛了些。不过话说回来，把这两个定义作这种比较也不太公平，因为语言的第二个定义本身就是对形式语言下的，自然是偏重语言的形式方面的抽象。而且，这一定义有两个明显的好处：

1．这个定义的确是覆盖了任何语言。从最具人为色彩的二进制数的语言到最复杂的自然语言，例如汉语，都被网罗到这一概念之下。二进制数语言不用说了，汉语符合这一定义也容易看出，只要取字母表V为所有汉字（或许还要加上标点符号）的集合，则（书面）汉语显然是V^*的一个子集。不言而喻，本书的主要讨论对象程序设计语言也被这一语言定义所包容。

2．这一定义足够地形式化，足够地精确，足够作为我们以后研究语言的基础，使我们能在这一严格的基础上发展

出一批知识。

在以上讨论的基础上，我们就可以明确确定本书对程序设计语言的研究所采取的立场了。这就是：以语言是语法和语义的统一体这一基本事实为原则，把程序设计语言看成字母表上句子的集合而将其纳入更一般的形式语言范畴之内，以便运用形式语言理论的结果处理程序设计语言的语法问题，进而在语法研究的基础上讨论其语义。

下面来谈语言的语法和语义。在阐述这两个概念时，摘引一下 Morris 对语言三要素的定义是很有益的。

符号语言学的创始人，美国哲学家 Charle Morris 把对符号语言的研究分为语法、语义、语用三部分。在他的《符号、语言和行为》一书中，Morris 是这样定义语言的三个要素的：

语用——关于在某种行为中发生的语言符号或信号的由来、使用和效果。

语义——关于语言符号或信号在其所有表现方式中的意义。

语法——有关语言符号或信号的组合方式而不涉及这些符号或信号所表达的意义，也不涉及这些符号或信号与它们发生于其中的行为之间的关系。

这种明确地把语言划分为语法、语义、语用三个要素，通过对它们的分别研究来把握语言整体的方法已被大多数语言学者奉为圭臬。应当说，Morris 的语言三要素抽象也确实是客观地反映了现实中各种语言的属性。以下我们就来分别讨论语言的这几个要素。

语法

与汉语的“语法”这个词相对应的英语单词有两个：“syntax”和“grammar”。一般情况是，对于自然语言的语法，我们使用grammar，而在谈论程序设计语言这类人为语言时就使用syntax。不过应当指出的是，对于计算机语言学者，syntax和grammar还有另一层细致的区别。grammar通常用来指控制语言中行的产生的规则，而syntax通常被看成计算机用以辨识一给定行是否为语言中合法行（有时我们把这种合法行称为语言的一个实例）的规则。这个问题，以后我们还有机会作更深入的讨论。这里我们使用的语法（syntax）一词就是笼统地指语言（特别是程序设计语言）的形式。而把grammar译为“文法”，意指形式语言学中的形式文法。

语法，即语言的形式，它表现为字符以及由字符构成更大的语法单位（如句子）的规则。根据这些规则就可以辨识出语言的哪些构成形式是语法上可接受的。换句话说，语法决定了一语言的一切合法构成。

设如我们有这样的一种“汉语”，其中有这样的两条语法规则：

“一个

名词　动词　名词

的序列就是一个句子”。　　(1 a)

“字符

人

饭

是名词，

吃

是动词”。 (1 b)

那么，

人 吃 饭

饭 吃 人

均为这种汉语的合法句子。尽管我们知道，只有第一句才于理相合而第二句子是有背情理的，但这里我们讲的是语法，只讲形式而不管意义。

第二个例子是前面已经提到过的二进制数语言。可以简单地给这个语言作如下的语法定义：

“字符 0 和 1 的一个有限序列构成句子。”于是

0

1

101

10101

等都是二进制数语言中的合法句子。至于 101 和 10101 究竟是什么东西，我们暂时全不知晓。

总而言之，应当始终记住的是，决定一语言结构是否能合法地接受的唯一判断标准是语法规则。

关于语法上正确的某种语言构成形式(比方说，句子)，其意义可以从两种角度来看。这就是，说（或写）句子的人意图表达的意义与听（或读）句子的人所理解的意义。前者就是语言的语义（semantics），后者就是语言的语用（pragmatics)。大多数情况下这两种意义总是相同的，否则的话，人们就不能利用语言来有效地交流思想了。但这两种意义又并非总是相同的，于是乎才可能有误解发生。在幽默作品和相声演出中，可以找到故意使语义和语用意义不一致从而闹

出笑话的大量例子。由此可知，把语义和语用区分开来对待是恰当的。下面让我们先来看看语义。

语义

简单地说，语义就是与语言的形式相联系的意义。既然如此，在规定某语言成分的意义时也就绝对离不开该成分的构成规则。换言之，规定语义时必须以语法作为基础。例如对于上述“汉语”一例中的句子

人 吃 饭

必须要在语法规则(1 a)和(1 b)的基础上来规定其语义。

根据(1 b)规定名词代表客体，动词代表动作。

根据(1 a)规定前一名词代表的客体把动词代表的动作作用于后一名词代表的客体。

这样一来，句子

人 吃 饭

的意义也就确定了。

关于二进制数语言的例子，情况也一样。对这种数的通常解释是：每个二进制数表示一个自然数。例如

101　　表示自然数 5

10101　表示自然数21

通过这种使语言的每个成分对应于一个熟知的数学对象的方法来定义语言的语义。在这里，如何具体对每个二进制数，确定出它所表示的自然数，也就是说如何对整个语言，而不是对语言的某特定结构给出其语义定义，那又必须借助于语法规则了。在以后的讨论进程中，我们还会回到这个问题。

语用

关于语用，前面已经说过，它可以简单地看成语言的接

收者所理解的意义。由此可以推想，对于自然语言和程序设计语言这样的人为语言，语用的影响无疑会有所不同。今针对我们的主要研究对象程序设计语言来讨论一下这个问题。一方面，程序设计语言具有人与人之间交流算法等方面的目的，因而有可能发生写程序的人意欲表达的意义被读程序的人误解这种情况。语义和语用的这种不一致可以通过语言的标准化、精确的语法和语义定义，以及正确的程序设计逻辑和良好的程序设计风格（主要指使用记忆标识符、模块化、完整的内部程序文件等用以提高程序可读性的技术）来排除。另一方面，也是更主要的一方面，程序设计语言又是人与计算机通讯的工具，是要被计算机所理解的语言，那么，这种情况下的语用影响又表现在什么地方呢？计算机是死的机器，它对语言的理解方式实际上是人强加给它的，因此我们总可以使计算机对语言的理解与语言本身表达的意义相吻合。换句话说，对于作为人机通讯工具的程序设计语言，我们总是可以作出安排，使语义和语用没有什么差别。但随之而来的是，要怎样才能办到这一点呢？显而易见，这已经完全是如何实现语言的问题了。从以上讨论可知，正如Morris对语用下的定义那样，“关于在某种行为中发生的语言符号或信号的由来，使用和效果”，程序设计语言的语用牵涉到语言的实现技术、程序设计方法学，以及语言开发和使用的历史诸多方面。由于对构成语用的这些问题的讨论是计算机科学中专门的研究领域，是需要大部头专著论述的课题，因此本书基本上不涉及语用学。只是在讨论语言的语义时，于必要之处注意一下语用的影响。比如，我们在讨论二进制数语言时，无论语法和语义定义得如何严格，但要实现

这个语言，把二进制数表示于定长存贮器之中时，我们就不得不考虑到语用的影响了。因为，由于存贮器容量的限制，是绝不可能对此语言中的任意数照语法和语义所规定的那样实现的。

在结束本节之前，我们打算略略提一下语言的二义性问题。对于这个程序设计语言的重要问题，后文将更详细地讨论。

二义性

所谓语言的二义性，顾名思义，就是可以把同一个语言表示解释成二个以上的含义。语言的二义性可以由语法的不适当定义引起，可以由语义的不适当定义引起，也可以由语义和语用意义间的混淆引起。

自然语言中充满着数不清的二义性语言表示。特别是，在口语中，语义和语用意义常有差别。重读、语气的变化等都会导致二义性。在广告标牌等简缩使用语言的场合，也常常可以见到有二义性的语句。例如，在飞机上就钉有写着如下英语语句的牌子：

NO SMOKING AREA IN REAR CABIN

这句话究竟意味着

后舱中有一不允许吸烟的区域，　　(1 c)

还是指

在后舱中没有允许吸烟的地方　　(1 d)

呢？这一语句的二义性显然是由于没有正确地使用连字符引起的。如果要把该语句解释成(1 c)则应写为，NO-SMOKING，如果要解释为 (1 d)，则应写为NO SMOKIND-AREA。这是语法二义性的一个例子。