

文章编号: 1672-5913(2011)11-0037-08

中图分类号: G642

文献标识码: A

关注编译原理的等价性

赵银亮

(西安交通大学 计算机系, 陕西 西安 710049)

摘 要: 本文对编译原理的等价性进行了分析和研究, 提出一种有效的组织等价知识的方法, 并在此基础上提出应用等价关系提高教学效果的一种方法论。结合编译原理教学实践说明了该方法的可用性和有效性。

关键词: 知识等价性; 教学方法论; 教学效果; 编译原理

1 背景

在教学中, 我们常常会使用断言: “甲与乙是一样的”, “甲乙二者是等价的”等等。从教和学的角度, 这样的断言有如下作用:

1) 建立知识之间的联系, 有助于对知识进行组织, 构建知识框架;

2) 减轻记忆的负担, 即本来对“甲”和“乙”的细节需要分别地记忆, 现在二者被等价关联起来, 有大量细节不必重复记忆了;

3) 便于知识的灵活运用。例如原来使用“甲”知识解决的问题换做用“乙”知识去求解或许是一条更加简洁有效的途径; 另外, 求解某问题时, 可用的知识从原来的一种增加到两种, 理论上增加了求解成功的可能性;

4) 便于突出重点, 对于等价的知识, 可以从其他的适合性角度做出选择, 不使总的知识量过于膨胀。

从以上可以看出, 这种教学现象反映了教学过程中对知识等价性的关注。显然, 关注知识等价性的教学过程对于提高教学效果必定产生积极作用。

等效思维、等价转化、等值变换等一些思维方法在教学和研究中被经常使用, 从一定程度上说明关注知识等价性的教学过程已经广为人们所接受。然而, 不幸的是, 这些思维方法仅停留在概念层面, 属于哲

学上的思考, 有模糊性, 尚没有形成系统的方法论(methodology)。事实上, 由于这一原因, 在课程教学中, 有效地运用这种关注知识等价性的教学方法面临着实质性的困难。

编译原理是一门传授对计算机程序进行分析、变换和优化的课程, 是计算机专业核心课程, 课程的知识分布在理论、技术和实现等多个层面, 多年来的教学实践反映出该课程是本专业学生最难学的课程之一, 因而, 结合该门课程的教学, 系统地研究关注知识等价性的教学方法有典型性, 同时从提高本门课程教学效果上也方便验证该方法的有效性。

本文提出了一种关注知识的等价性的教学方法, 并应用于编译原理课程教学中, 给出了组织等价知识的方法, 以及提高教学效果的形式规则, 并基于编译原理教学实践阐明了该方法的有效性。本文后面部分: 第二节分析等价性的利用对于提高教学效果的积极作用; 第三节提出一种关注知识等价性的方法论; 第四节结合编译原理课程分析了该方法的可用性和有效性; 最后是本文总结。

2 等价性的利用对提高教学效果的意义

一般来说, 恰当地使用知识之间的等价关系, 对于提高教学效果产生积极作用。等效思维、等价转化以及等值变换等这些思维方法, 是以知识的等价性为

基金项目: 国家 863 课题“面向 CMP 的推测多线程编译技术研究”(2008AA01Z136)。

作者简介: 赵银亮, 男, 教授, 研究方向为语言及编译系统, 并行计算与机器学习。

基础所建立起来的,在教学过程中有意识或无意识地得到了广泛使用。本节在介绍这些思维方法的基础上,结合编译原理中活动记录(栈帧)知识对这些方法进行了具体解释,并基于此进行了初步分析。

2.1 基于等效思维举一反三

所谓等效思维,就是在效果相同的前提下,把实际的、难于理解的问题变成理想的、易于思考的问题。它是数学上“等价转化思想”在科学学科的问题求解中的应用,是一种常用的有效的科学思维方法。它常常是通过变换思维角度,对题目作适当假设,或者代换。

活动记录是一种数据结构,用于运行时栈空间的管理,目的是支持函数调用的语义。活动记录的结构有多种,常见的是 $\text{fp-sp}^{[1]365}$ 和 $\text{sp-top}^{[2]259}$ 两种结构。这两种结构中的内容是完全一样的,也就是效果相同,但也存在着差异。 fp-sp 结构的特点是,帧指针 fp 把活动记录分为两个部分,前一部分的内容由调用者中的代码来建立,而后一部分的内容由被调用者中的代码建立。 sp-top 结构中帧指针 sp 指向活动记录的开始位置,这样帧与帧之间的边界清楚。但是访问 sp-top 结构的单元所使用的偏移量的绝对值要大。基于等效思维,抛开了这些差异,从 fp-sp 结构容易得出 sp-top 结构,反之亦然。进一步地,容易设计出不同的活动记录结构,以及相应的调用序列(calling sequence)和返回序列代码,取得举一反三的效果。

然而,在这一思维方法中,对于“实际的、难于理解的”知识和“理想的、易于思考的”知识,缺乏方法论层面的定义和解释,而“效果相同”也是泛化和不明确的陈述,因而,可以看出这一思维方法仍然停留在概念层面上,缺乏形式化描述,这成为有效应用这一方法的瓶颈。

2.2 基于等价转化化难为易

在数学研究中,使一种研究对象在一定条件下转化为另一种研究对象的数学思想称为等价转化思想,是一种思维策略的体现,即我们常说的换个角度想问题,它是解决数学问题的重要思想,它要求我们能把握住问题的本质,能辨证地看待事物,能运用所学的

知识把复杂的问题转换为较简单的问题解决,把隐含的条件转化为明显的条件,把生疏的问题转化为较熟知的问题去解决。

运行时系统常常基于栈帧结构来支持嵌套过程的调用和返回。采用 D 表(Display 表)的栈帧结构提高了时间效率,但理解起来略显繁琐。与它等效的采用静态链的栈帧结构理解起来更简洁一些。除此以外,二者没有区别。因此,基于对静态链栈帧结构理解的基础上,通过将静态链等价转化为 D 表,使得后者化难为易了。

然而,在这一思维方法中,对于“一定条件”和“研究对象”缺乏形式化描述也成为有效使用该方法的障碍。

2.3 基于等值变换法进行创新思维

等值变换法是从已有的事物中,通过模拟、借鉴、产生联想来改变原来的对象而进行创造的方法,是依照蚕变成飞蛾、桑蚕变成蚕丝的自然现象而总结出来的一种创造技法。自然界的各种等值变换形式归结为三种类型:1)自我成长型等值变换,即类似于蚕从幼虫变化为成虫的变换过程;2)被加工型等值变换,即类似于从桑叶到蚕丝这一变换过程;3)综合型等值变换,即前两种的综合。

仍然以栈帧知识为例,我们能注意到,嵌套过程的语义是一种知识,而栈帧结构及其调用序列和返回序列构成了另一种知识,在一定的抽象层次上看,这两种知识之间存在着等价性。基于等值变换这种思维方法,后一种知识是前一种知识经第二类等值变换得到。进一步地,我们可以给栈帧结构中增加表示超长、变长形参单元,可以支持过程闭包和变长参数语义等等,依次可以看出是对栈帧知识的第一类等值变换。等值变换使得编译原理的理论知识转变为方法层面的知识,并最终转变为程序代码,就是组成编译系统的代码。

等值变换法的核心是保持等值,从已有到未知。目前仍然没有形式化描述,没有适用于计算系统的方法论。

综上所述,基于知识的等价性,将等效思维、等

价转化以及等值变换等思维方法应用在编译原理教学中,必定取得举一反三、化难为易和创新思维的教学效果。但是,教授和学习每一种涉及到等价性的知识的过程中,准确描述等价关系是决定能够利用等价性提高教学效果的前提,这将在下一节具体阐述。除此以外,基于等价性剪裁教学内容的意义在于:将适应不同的学分要求、适应不同的学生基础这件事,转变成为在多种等价知识中进行选择的问题,显然都变得易于实施。

3 关注知识等价性的教学方法论

本节采用一种层次结构组织知识,并定义知识间的等价关系,在此基础上引入知识等价性的层次视觉和抽象视觉,给出等效思维、等价转化和等值变换等思维方法的一种形式化描述,形成一种关注知识等价性的教学方法论。

3.1 知识单元及其层次结构组织

知识是哲学层面的概念,为了应用于计算系统中并得到方便处理,采用知识单元的概念。

定义 1 知识单元通常是指在一个问题域中用于阐述或解决某个问题的概念、算法、定理等,它在内容上是相对独立和完整的。依据是将知识作为某个对象看待,作为结构化的知识要素,作为知识内容的不可再分的打包形式。知识单元有时采用集合形式表示,其元素是知识个体或称为知识实例。

将编译原理的内容作为一个问题领域,记为 C ,那么 C 是编译原理的知识单元的全集。知识单元不是孤立的,它们之间存在着大量的联系,我们把它表示成知识单元之间的关系。知识抽象是知识单元之间的一种常见的关系,知识抽象关系的全集记为 A 。那么有:

(1) $\Rightarrow \forall (\alpha: C \rightarrow C) \in A, \alpha$ 是 C 上严格偏序关系

(2) $\forall \alpha_1, \alpha_2 \in A, \forall K, K_1 \in C \rightarrow$

$(K, K_1) \in \alpha_1 \rightarrow (K_1, K) \notin \alpha_2, (K, K_1) \in \alpha_2 \rightarrow (K_1, K) \notin \alpha_1$

给定 $A \subseteq A$, 按照 A 把 C 的知识单元组织成为一个层次结构,当 $A=A$ 时,所得到的层次结构是完全的。在知识的层次结构组织中,上层知识单元是下层

知识单元的抽象。层次结构的底层是程序代码,顶层是特殊的知识单元。

编译原理中的知识单元划分为三个层次:理论层次、技术(方法)层次和实现层次。理论层次表现为:定义、性质、定理、理论工具、理论模型等,方法层次包括算法、数据结构、技术模型等,而实现层次是程序代码,指构成编译系统的那些程序代码。

在数量上,理论层次的知识单元少、实现层次上的知识单元多,而方法层次介于中间。实现层次上的知识单元是完成某些功能的程序代码,由于相同功能的程序可以很多,所以这个层次上的知识单元被认为有无限多个。不同层次的知识单元之间存在着联系,等价性不仅存在于各个层次内部也存在于层次之间。等价性的层次视觉不仅关注各个层次内部知识单元间的等价关系也关注层次之间知识单元之间的等价关系。

3.2 知识单元之间的等价关系

在知识层次结构中存在等价关系。知识单元之间的等价关系不同于集合论中集合上的等价关系。

定义 2 知识单元之间存在三类等价关系。

(1) 对于任意 $K \in C$, 若存在 ρ 是集合 K 上的等价关系(数学中的等价关系), 那么 $\gamma = \{(e, \rho(e)) | e \in K\}$ 是 K 与自身间的等价关系。等价关系 γ 总是存在的。

(2) 对于任意 $K_1, K_2 \in C$, 且 K_1 和 K_2 有共同的抽象父结点, 若存在一一对应映射 $\rho: K_1 \rightarrow K_2$, 那么 $\eta = \{(e, \rho(e)) | e \in K_1\} \cup \{(\rho^{-1}(e), e) | e \in K_2\}$ 是 K_1 与 K_2 之间的等价关系。注意, 任意两个有共同抽象父结点的知识单元之间, 在有意义的抽象层次上不一定存在等价关系。

(3) 对于任意 $K_1, K_2 \in C$, 且 K_1 是 K_2 的抽象父结点, 若存在映射 $\rho: K_2 \rightarrow K_1$, 那么 $\nu = \{(\rho(e), e) | e \in K_2\}$ 是 K_1 与 K_2 之间的等价关系。这个等价关系必定存在(把 K_2 到 K_1 的抽象关系作为 ρ)。

3.3 等价关系的抽象与等价性金字塔

C 中等价关系的全集记为 \mathcal{E} , 定义 \mathcal{E} 中元素间的抽象关系的全集为 \mathcal{B} , 如下:

(1) $\Rightarrow \forall (\beta: \mathcal{E} \rightarrow \mathcal{E}) \in \mathcal{B}, \beta$ 是 \mathcal{E} 上严格偏序关系

(2) $\forall \beta_1, \beta_2 \in \mathcal{B}, \forall R, R_1 \in \mathcal{E} \Rightarrow (R, R_1) \in \beta_1 \rightarrow (R_1, R) \notin \beta_2, (R, R_1) \in \beta_2 \rightarrow (R_1, R) \notin \beta_1$

给定 $B \subseteq \mathcal{B}$, 按照 B 把 \mathcal{E} 的等价关系组织成为一个层次结构, 称为等价性金字塔。当 $B = \mathcal{B}$ 时, 所得到的层次结构是完全的。上层等价关系是下层等价关系的抽象, 顶层是一个特殊的等价关系 R_0 , 定义为 $(R_0: \mathcal{C} \rightarrow \mathcal{C}) = \mathcal{C} \times \mathcal{C}$ 。

等价性金字塔也就是知识的等价性的抽象视觉, 揭示了等价知识单元之间等价性的具体体现。如图 1 所示, 其中 S, Q, R , 和 P 是不同抽象层次上的等价关系, 上下连线表示抽象, 上方是下方的抽象, 如 S 是 Q 的抽象。

等价性的抽象视角表达的是为什么知识单元等价, 具体来说是在某一抽象层次上看是等价的, 否则就不是等价的, 例如 $\exists S, (x, y) \in S \Rightarrow (x, y) \notin Q$ 。等价性金字塔描述了任意等价知识单元所在的抽象层次的范围, 越往塔上部, 表示抽象层次越高, 否则越低。如果两个知识单元在较高抽象层次上等价, 则它们之间的共性的东西少, 否则共性多, 例如 $S \subset Q \cap R$ 。最底层的等价只有共性没有个性。而最上层应该就是概念上的等价, 即编译原理中所有的知识单元都是等价的, 共性只有一个, 即它们都是编译原理中的知识单元。

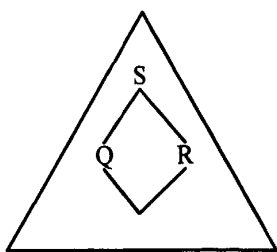


图 1 等价性金字塔示例

在等价性金字塔的某个位置, 若两个知识单元是等价的, 则在这个位置之上的其他位置, 这两个知识单元仍然是等价的, 即 $(x, y) \in Q \Rightarrow (x, y) \in S$ 。

等价性层次视觉和等价性抽象视觉给出了确定等价关系的方向。在给定抽象层次上, 两个知识单元之间是否能找到等价关系是判别这两个知识单元之间是否存在等价性与否的基准, 而等价关系还具体描述了

为什么两个等价知识单元是等价的。如果两个知识单元之间的等价关系不止一个, 那么这些等价关系的交集组成了一个更为抽象的等价关系, 在这个等价关系下, 原来的两个知识单元仍是等价的。

3.4 教学效果的形式化描述

定义 3 教学效果

(1) 等效思维的形式化描述

$\forall K \in \mathcal{C}, \exists (\gamma: K \rightarrow K) \in \mathcal{E}$, 给定 $a \in K$ 找出 $b \in [a]$,

其含义是从知识单元的一些知识实例出发能够找出等价的另一些知识实例, 体现了举一反三的效果。

(2) 等价转化的形式化描述

$\forall K_1, K_2 \in \mathcal{C}, \exists \alpha \in \mathcal{A}, \alpha(K_1) = \alpha(K_2), \exists (\eta: K_1 \rightarrow K_2) \in \mathcal{E}$, 给定 $a \in K_1$ 可以得出 $\eta(a) \in K_2$

其含义为在抽象层次上看相同的两个知识单元各有其特性, 二者之间的相互转化是等价转化, 二者的元素(知识个体)仍然可以相互转化, 有助于问题求解。

(3) 等值变换的形式化描述

$\forall K_1, K_2 \in \mathcal{C}, \exists \alpha \in \mathcal{A}, K_1 = \alpha(K_2), \exists (v: K_2 \rightarrow K_1) \in \mathcal{E}$, 给定 $a \in K_1$ 可以得出 $v^{-1}(a) \in K_2$

$\forall K_1, K_2 \in \mathcal{C}, \exists \alpha \in \mathcal{A}, K_1 = \alpha(K_2), \exists (v: K_2 \rightarrow K_1) \in \mathcal{E}$, 给定 $a \in K_2$ 可以得出 $v(a) \in K_1$

即从较抽象的等价类中的知识单元集合变换为较具体的等价类中的知识单元, 而最终的方向是到编译器程序代码。

4 关注等价性的教学实践

4.1 过程调用语义与栈帧的等价性

首先以栈帧为例, 栈帧可细分为多个栈帧知识单元, 如图 2 所示。图 2a 和图 2b 所示两种活动记录, 它们所含的内容相同, 只是排列布局不同。所以它们之间的等价关系就是同为活动记录所含内容相同。不同的排列布局导致生成的建立活动记录的代码不同。对于优化静态链方式的活动记录, 将静态链单元作为全局 D 表指针使用, 而具体 D 表单元添加在形参块外侧, 见图 2c, 得到等价的带 D 表栈帧。由于活动

记录中的参个(参数个数)单元限定了形参必须等长度,那么对于超长(如函数闭包)和变长(如变长数组)的参数,采用指针指向存放在临时单元中的这些对象,如图 2d 所示,临时单元添加到活动记录的靠栈顶指针那一端,得到变长参数栈帧和过程闭包栈帧。

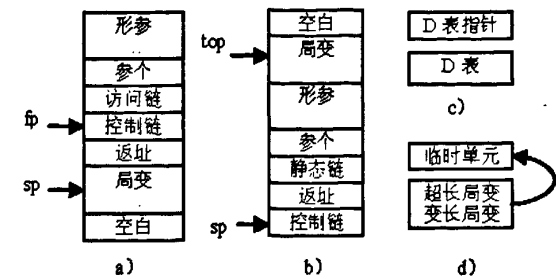


图 2 活动记录(栈帧)

过程调用语义可以细分为几个知识单元,如非嵌套过程、嵌套过程、变长参数、过程闭包等。分别对应不同的栈帧知识单元。图 3 给出了栈帧及相关知识单元,图中略去了各种不同语义知识单元。对应的栈帧为基本结构,可以等值变换为静态链栈帧和 D 表栈帧,进一步,可等值变换为变长参数栈帧和过程闭包栈帧,并最终等值变换为程序代码。

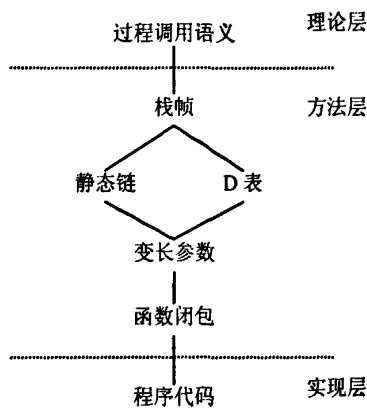


图 3 栈帧相关知识单元

在此基础上,基于定义 3,我们得到如下结果:

(1) 等效思维举一反三描述为,在各自知识单元内部(即保持栈帧内容不变),改变栈帧各单元的排列位置,并对应地编写调用序列代码和返回序列代码,分析其优缺点。

(2) 等价转化化难为易描述为,构造静态链栈帧和 D 表栈帧之间的等价转化方法;在过程调用的不同语义知识单元之间寻找等价关系,并基于等价性金字塔解释这些等价知识单元之间的共性和个性。

(3) 等效变换创新思维描述为一系列等值变换过程,自上而下地有:过程调用语义知识单元到栈帧知识单元的第二类等值变换,栈帧到其他栈帧的第一类等值变换,栈帧到程序代码的第二类等值变换。自下而上地,从下层知识单元抽象出方法层面上的栈帧,再次抽象出理论层面上的过程调用语义。

综上所述,运用关注知识等价性的教学方法论,将过程调用语义的各个知识单元及其间的等价关系形式化地表达出来,并能够确切实施等效思维、等价转化和等值变换等思维方法,达到举一反三、化难为易和创新思维的教学效果。

4.2 正规式、正规文法和有穷自动机的等价性

正规式、正规文法和有穷自动机的知识单元及抽象关系示意如图 4 所示。

理论层次上,正规式 r ,正规文法 G 和有穷自动机 M 之间的等价关系 S 可以用 $L(r)=L(G)=L(M)$ 来定义,即它们的语言相等就表示它们相互等价,进一步 S 可以转变为 Q 如下(Q 也是等价关系):

- (1) $s \in L(r)$ 当且仅当 s 与 r 进行模式匹配成功;
- (2) $s \in L(G)$ 当且仅当 s 由 G 的开始符号推导出,或者;
- (3) $s \in L(M)$ 当且仅当 s 由对 M 执行得到。

显然 Q 是一定抽象层次上的等价关系,但略去了这三者的如下差异:

- (1) s 与 r 的模式匹配是陈述性的, s 不是结构化的, r 是对 $L(r)$ 的概括(模式);
- (2) 由 G 的开始符号推导出 s 是过程性的,可能有回溯, s 是结构化的(结构是指非终结符);
- (3) 执行 M 得到 s ,这是过程性的,没有回溯,便于程序实现, s 不是结构化的。

在较具体的层次上,把这三者相互等价转换的方法记为等价关系 R 。具体如下:

- (1) r 与 G 等价转换。通过命名,引入正规定义,将 r 中的闭包运算消除掉,就得到 G ;将 G 中的产生

式规则相互代入,当只剩下一个产生式规则并且左部为开始符号时,右部就是 r 。

(2) r 与 M 等价转换。构造 NFA, 开始状态到结束状态弧上标记为 r , 等价变换为不含闭包和选择操作的字符串作为弧标记的 NFA, 再等价转换为 DFA。反之给定 NFA 或 DFA, 进行等价转换, 得到只有一个开始状态和一个结束状态的 NFA, 之间弧上标记即为 r 。

(3) M 与 G 等价转换。 G 为右线性文法或左线性文法, 可直接构造出 NFA。给定 DFA, 直接构造出 G 。

实现层的知识单元是程序代码, 有基于模式匹配的、基于文法推导的、基于状态转换的三种词法分析程序, 如图 4 所示, 分别为 *matcher*, *scanner*, *parser*。实现层的等价关系 P , 是程序之间的等价关系, 即如果它们接受的串的集合是同一个的话。

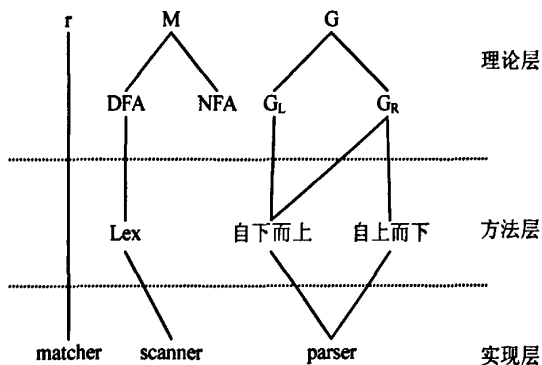


图 4 正规式、正规文法、有穷自动机知识单元

总之, 这三者从不同的层面(个性不同)表示了语言以及如何判断一个符号串是给定语言的句子的方法。不强调结构是词法分析的特点, 但就基于文法的词法分析也像语法分析一样有语法树, 只是内结点没有意义罢了。当然附带意义的也有, 比如词法分析中计算数值 *token* 的值。基于此可以拓展思维, 有助于灵活掌握知识。

在此基础上, 基于定义 3, 我们得到如下结果:

(1) 等效思维举一反三可以描述为任意给定一个正规式(文法和自动机), 列举出与它等价的另一个; 进一步可考虑这些等价的知识个体有什么不同特

点, 比如 DFA 化简实质上是等价的 DFA 中选出状态数最少者。对于等价的正规式, 则有表示形式上的繁简及喜好之别。对于等价的文法而言, 差别在于是不是 3 型文法、有无左递归、回溯等。

(2) 等价转化化难为易可以描述为给定 NFA 等价转换为 DFA; 左、右线性文法 G_L 和 G_R 相互转换; 针对 G_R 的自上而下和自下而上分析的等价转换等。

(3) 等值变换创新思维可以描述为沿着图 4 中从上往下的路径, 将较抽象的知识单元等值变换为较具体的知识单元, 或者反之。从上往下的等值变换对应于自然的演化过程, 最终演化成程序代码, 体现了理论原理指导编译器实现的过程。另一方面, 从较具体的知识单元中寻找较抽象的等价知识单元, 反映了对规律性的认识。即, 词法分析程序的规律性就是 r 或 M 或 G 。

4.3 属性文法、翻译模式等价性

属性文法^{[2]136}, 又称为语法制导的翻译模式(缩写为 SDT)^{[3]195}, 是进行语义分析的理论工具, 有三种类型:

(1) SDT: 将{语义动作}出现在候选式中, 形如 $A \rightarrow \{1\}\alpha\{2\}\beta\{3\}$, 其中{1}、{2}、{3}表示语义动作(或称语义规则、方程);

(2) 后缀式 SDT: 引入 ϵ 产生式, 替代候选式中的 { } 出现, 形如 $A \rightarrow M_1\alpha M_2\beta\{3\}$; $M_1 \rightarrow \epsilon\{1\}$; $M_2 \rightarrow \epsilon\{2\}$;

(3) 属性文法(AG): 产生式规则附加语义方程, 形如 $M_1 \rightarrow \epsilon\{1\}$; $A_1 \rightarrow M_1\alpha\{2\}$; $A \rightarrow A_1\beta\{3\}$;

如图 5 所示, 在理论层面上, 我们只需考虑使上述三种翻译模式与语法分析配合起来完成语义分析。理论层面上通过文法修剪, 非终结符替换可以实现这三者等价转换, 定义为等价关系 R 。

为了简单起见, 我们不考虑候选式最左边出现语义动作, 以及 ϵ 产生式的左部非终结符的情况, 即上述三种情况中均不存在语义动作 {1} 的情况。这时与语法分析集成的工作就简单了, 即对于 LR 分析, 符号栈栈顶出现 α 时立即执行 {2}, 出现 β 时立即执行 {3}。

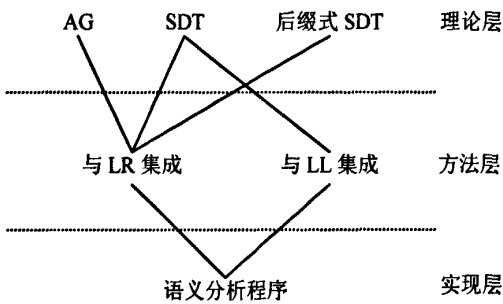


图 5 属性文法与翻译模式知识单元

在这种简化了的情况下，仍然通过文法修剪和非终结符替换可以实现这三者等价转换，定义为等价关系 S。显然 S 是 R 的抽象。

在技术层面上，要具体化如何与 LR 分析的集成。对于 AG 和后缀式 SDT，归约时执行语义动作，而对于 SDT，符号栈顶发现 α 或 β 即进行相应语义动作，把这样的转换定义为等价关系 Q。这三者的差异表现在：实现 SDT 的动作要比实现 AG 和后缀式 SDT 的动作更困难；后缀式 SDT 复杂化了候选式(增加了长度)容易导致语法分析产生组合爆炸问题。在技术层面上，等价关系 Q 是在 S 的基础上考虑了与 LR 的具体集成，而等价关系 P 是在 R 的基础上考虑了与 LR 的具体集成(集成方式略去)。这些等价关系之间的抽象关系如图 1 所示。

在此基础上，基于定义 3，我们得到如下结果：

- (1) 等效思维举一反三可以描述为寻找等价的 AG(SDT 或后缀式 SDT)，差别是采用的属性不同(文法符号的不同属性、属性值的不同计算方法、对翻译结果的不同表示，对翻译结果的优化等)，或者对语法做不同修剪。
- (2) 等价转化化难为易可以描述为 AG、SDT 和后缀式 SDT 之间等价转化，差别是对文法修剪的程度、对左递归和回溯的处理、非终结符有无语义意义等。
- (3) 等值变换创新思维可以描述为理论层的三种知识单元如何集成到语法分析模型中，以及进一步实现在语义分析程序中。

基于此，一种结果是，在重点介绍 AG 及基于

AG 的语句翻译或中间代码生成的基础上，基于 S 简要引入 SDT 或后缀式 SDT，比直接介绍 SDT 的方式^{[3]195}易于为学生理解和接受。另一种结果是，同样在重点介绍 AG 的基础上，基于 R，简要介绍 SDT，能达到同样的效果。以上作为强调理论性的课程^[4]内容选择。如果基于 Q 或 P 考虑，重点放到 AG 以及 LR 分析的集成上，但比前者稍难。另外，在实现层面，语义分析程序基于三种翻译模式都可以工作，它们之间的等价性过于陷入实现细节，是强调实践性的课程^[4]内容选择。

4.4 名字作用域与层次符号表等价性

图 6a 表示名字作用域语义，其中名字可以是形参、局变、过程等。就过程 Q 而言，其中定义的名字 x(用 x:表示定义 x)词法上在它的后面可以被访问(用:x 表示引用 x)。Q 的外层过程中位于 Q 定义之前定义的名字比如 y 可以在 Q 中访问(图中用:y 表示)。那么图 6b 所示的层次符号表结构和符号表操作合起来(是一个抽象数据类型)等价地表示了此语义。如图 6b 所示，上部为 P 的符号表，下部为 Q 的符号表，当对 Q 进行语义分析时，对: x 的处理只需查找 Q 符号表即可，而对:y 的处理首先查找 Q 符号表，查不到的情况下再沿上层链找到 P 符号表中，即可查到。如果上层链为空，则表示出错(该名字未定义)。可以看出以上对应第二类等值变换方法。

总之，知识单元(如图 6a 所示)等值变换为知识单元(如图 6b 所示)。理论层面的知识单元(如图 6a 所示)等值变换为技术层面的知识单元(如图 6b 所示)。

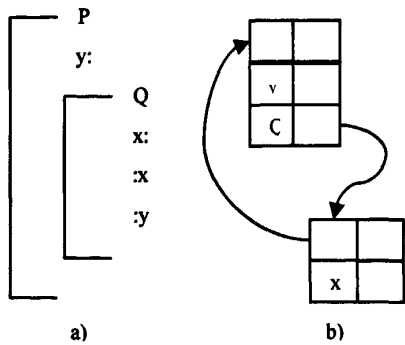


图 6 作用域知识单元

5 结语

在教学中,基于知识抽象的知识单元层次结构能够用于有效组织编译原理课程中的理论、方法和实现层面上的知识,而知识单元之间存在的等价关系抽象为等价性金字塔表示,可以解释任意知识在一定抽象

层次之上都存在等价关系。在此基础上提出的关注知识等价性的教学方法论,确定性地表达了举一反三、化难为易、创新思维等教学效果。已有的教学实践反映出这种等价教学法的可用性和有效性。进一步工作是在课程实践中归纳出更多的等价关系并进行性能评估。

参考文献:

- [1] Kenneth C. Louden. Compiler Construction: Principles and Practice[M]. 机械工业出版社, 2002: 582.
- [2] 陈火旺, 刘春林, 谭庆平, 等. 程序设计语言编译原理[M]. 3版. 北京: 国防工业出版社, 2001.
- [3] A. V. Aho, M. S. Lam, R. Sethi, ed al. Compilers: Principles, Techniques, and Tools[M]. 2nd ed. 赵建华, 等译. 北京: 机械工业出版社, 2009.
- [4] 赵银亮. 浅谈编译原理课程的定位[J]. 西安邮电学院学报, 2010, 15(4): 143-147.


Concerning about the Equivalence of Compiler Theory

ZHAO Yinliang

(Department of Computer Science, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: This paper proposes an effective method for organizing equivalent knowledge. Based on the method this paper also proposes a methodology of improving teaching effectiveness by the application of this equivalence relation. And the teaching practice combination of compiler theory illustrates the availability and effectiveness of this method.

Key words: knowledge equivalence; teaching methodology; teaching effectiveness; compiler theory

(编辑: 白杰) 

(上接 36 页)


Analysis of the Knowledge Points of Compiler Principle in Multicore Course

YANG Jianfeng, SUI Zhucui, XIE Yinbo

(School of Electronic and Information, Wuhan University, Wuhan 430072, China)

Abstract: Multicore technology brings great changes of different fields in computer technology such as compilers. The principal key is which knowledge points should be selected in multicore course. In this paper, the knowledge points in multicore course are analysis and the requirements from the setting of the teaching content based on the "multicore architecture and programming technology" of Wuhan University according to the development of multicore technology and teaching experience in this course

Key words: multicore course; compiler; knowledge points

(编辑: 张玥) 

关注编译原理的等价性

作者: [赵银亮, ZHAO Yinliang](#)
作者单位: [西安交通大学计算机系, 陕西西安, 710049](#)
刊名: [计算机教育](#)
英文刊名: [COMPUTER EDUCATION](#)
年, 卷(期): 2011(11)

参考文献(4条)

1. [赵银亮](#) [浅谈编译原理课程的定位](#) 2010(04)
2. [A.V.Aho;M.S.Lain;R.Sethi;et al. 赵建华](#) [Compilers:Principles,Techniques,and Tools](#) 2009
3. [陈火旺;刘春林;谭庆平](#) [程序设计语言编译原理](#) 2001
4. [Kenneth C.Louden](#) [Compiler Construction:Principles and Practice](#) 2002

本文链接: http://d.g.wanfangdata.com.cn/Periodical_jsjjy201111010.aspx