

## 类比推理中的一种相似性准则 及其验证方法

张少平 王怀民 陈火旺  
(计算机系)

**摘要** 类比推理是人工智能领域中的重要研究方向,而相似性准则是类比推理中的重要概念。文中给出了类比推理的一种描述框架,定义了相似性准则,并且基于项重写技术,提出了相应的验证方法。

**关键词** 人工智能,推理,规范/类比推理,相似性准则,项重写 软件重用

**分类号** TP18

在人工智能领域中,类比推理已有很长的历史。类比是人们进行思维的重要方法之一,在理解、学习和概括等智能活动中都离不开类比。

使用类比推理大致上基于两个方面的原因:其一是根据我们已有的知识,无法解决某一问题,通过类比推理可以找出该问题的相似问题,然后,根据相似问题的解,作出关于该问题的一些新的猜想,从而解决问题;另一原因是依据我们所掌握的知识,不使用类比推理,虽可以解决问题,但使用类比推理可以快速、有效地解决问题,类比推理作为有效地获取和使用知识的途径。

在进行类比推理时,必须具备两个先决条件,

(1) 相似性准则。相似性准则是指两个问题相似的标准,即何时两个问题,针对某一目的而言是相似的。相似性准则对于类比推理是极为重要的,因为没有类比的标准,就无法进行比较。

(2) 源知识域。源知识域是指通过问题求解或其它方式所获取的知识全体。对于源知识域中的每一个问题,都有一个精确的解。

可以认为,类比是有效地联接推理者的过去和现在经历的基本机制。如果没有过去的经历和日积月累所掌握的源知识,类比就缺乏必要的物质基础。源知识域中的源知识的描述以及组织方法,直接影响类比推理的效果。

在本文中,以软件重用问题为背景,讨论了一类相似性准则问题。在定义这种相似性准则及其验证方法之前,让我们首先给出类比推理的一种描述框架。

## 1 类比推理的一种描述框架

对于类比推理,可以用一种简单的描述框架,描述其推理的全过程。该框架由源知识的表示与组织,源知识的识别与修改完善三部分组成。

### (1) 源知识的表示与组织

对于源知识,我们应以某种语言对其进行描述,并且对其加以分类,把它们有效地组织成为一个源知识域。

源知识域的组织是精确地验证相似性准则的基础,也是有效地识别源知识的基础。

### (2) 源知识的识别

根据相似性准则,验证识别出的源知识的有效性。

对于不熟悉其解的问题,我们如何把该问题与源知识域中的一个或多个源知识联系起来呢?最常见的方法就是根据相似性准则及其验证方法在源知识域上进行检索。

从源知识域中识别出的源知识与欲求解问题之间是否满足相似性准则,需要进行相似验证。验证可以分成语法验证和语义验证两部分。如果验证的结果是满足相似性准则,则称相应的源知识为相似源。

### (3) 修改与完善

因为相似问题的解是相似的,故对相似源的解稍作修改,即可得到关于欲求解问题的解。在得到该问题的解以后,还需要把该问题及其解存入到源知识域中,为将来的类比推理作好准备。

源知识域应当是开放型的,随着类比推理的逐步深入,以及其它方式的学习,源知识域应当逐步丰富与完善。

## 2 一种关于软件重用的相似性准则

软件重用的基本出发点是利用已实现的成熟软构件构造新的软件。基于重用的软件开发思想是对上述类比推理一般框架的自然的具体表现。

软件重用技术的物质基础是软构件库的建立与组织。软构件库是软件重用技术的源知识域。欲设计的软件与重用软件间的匹配原则是软件重用技术的相似性准则。与类比推理的一般框架相对应,一个软件重用检索系统由三个主要成份构成,即:重用软件库,软构件识别系统和软件修改完善系统。软件重用检索系统与问题分解系统和软构件组装系统结合在一起,构成了基于重用技术的软件开发系统的主干。简单地说,软件重用技术是依据我们已掌握的重用软件,快速有效地开发软件的一种类比推理技术。

类比推理首先必须解决的关键性问题是欲求解的问题与源知识的相似性原则,精确地规定相似性准则的基础是在统一的语义论域上对欲求解的问题和源知识进行形式化规范描述。

不同的类比推理系统,依据不同的相似性准则,因为相似性准则与论域紧密相关。由于我们所感兴趣的是利用类比推理进行软件开发,故用形式化的代数规范描述问题和重用软件<sup>[6]</sup>,并且基于代数规范定义相似性准则。

代数规范的一般描述为一个二元组:

$$sp = (\Sigma, E)$$

其中,  $\Sigma$  为代数规范的标记,  $E$  为等式公理, 它们分别对应问题的语法、语义描述。

假设源知识域中的成份为软构件, 软构件由其完整代数规范  $sp$  和该问题的一种正确有效的解  $code$  组成。现在的问题是, 若  $sp_0$  为欲求解问题的 (不完整) 代数规范, 何谓  $sp_0$  与  $sp$  相似呢? 直观上讲, 两个规范应具有相同的语法结构, 更重要的是两个规范所决定的初始模型也应当相似, 精确地说,  $sp_0$  中的等式公理, 应在  $sp$  所决定的初始模型下有效。可以对相似性准则形式定义如下:

**定义** 设  $sp_0$  为欲求解问题的规范 (不完整的或部分的),  $sp$  为源知识域中某个软构件的规范, 我们说  $sp_0$  与  $sp$  相似, 且记作:  $sp_0 \sim sp$  是指:

$$(a) \quad \Sigma_0 \subseteq \Sigma$$

$$(b) \quad \forall t = s \in E_0 \implies I_{sp} \models t = s$$

其中  $\Sigma_0 \subseteq \Sigma$  表示  $\Sigma_0$  是  $\Sigma$  的子标记,  $I_{sp}$  表示由规范  $sp$  所确定的初始模型,  $I_{sp} \models t = s$  是指等式  $t = s$  的初始有效性。

例如, 设表的代数规范  $sp_{l, l, l}$  中的等式公理为:

$$\text{append}(\text{nil}, l) = l \quad (1)$$

$$\text{append}(x \cdot l, m) = x \cdot \text{append}(l, m) \quad (2)$$

$$\text{rev}(\text{nil}) = \text{nil} \quad (3)$$

$$\text{rev}(x \cdot l) = \text{append}(\text{rev}(l), x \cdot \text{nil}) \quad (4)$$

对于一个关于表的欲求解的问题, 某用户只知道其问题的描述为  $sp_{l, l, l}^0$ :

$$\text{append}(x \cdot l, m) = x \cdot \text{append}(l, m) \quad (5)$$

$$\text{rev}(\text{rev}(l)) = l \quad (6)$$

可以验证:

$$I_{sp_{l, l, l}} \models \text{append}(x \cdot l, m) = x \cdot \text{append}(l, m)$$

$$I_{sp_{l, l, l}} \models \text{rev}(\text{rev}(l)) = l$$

从而验证了  $sp_{l, l, l} \sim sp_{l, l, l}^0$ , 这时可用  $sp_{l, l, l}$  的实现, 求解该问题。

### 3 相似验证

对于前面提出的相似性准则, 如何进行语义验证, 即证明初始有效性呢? 过去人们经常用的证明方法是基于某种结构的归纳法。这种方法的缺点是难以算法实现。八十年代, 人们受项重写技术中完全过程 (completeness procedure) 的启发, 发现了一种初始有效性的证明过程<sup>[11], [12]</sup>。这里我们仅仅介绍其证明过程的主要思想。

首先, 将代数规范的标记集族进一步地分解成构造符 (constructor) 和定义函数符。例如,  $sp_{l, l, l}$  的标记集族由构造符集族  $\{\text{nil}, \cdot\}$  和定义函数符集族  $\{\text{append}, \text{rev}\}$  组成, 一般情况下, 把前者记为  $C$ , 后者记为  $D$ 。称  $sp = (C \cup D, E)$  关于  $C$  是完全的是指  $sp$  满足:

(1) 对于由  $C \cup D$  生成的所有合法项  $t$ , 都存在由  $C$  生成的项  $s$ , 使得

$$t =_E s$$

即:  $t$  与  $s$  在等式理论  $E$  上相等

(2) 对于由  $C$  生成的项  $t, s$ , 若  $t =_E s$ , 则  $t$  与  $s$  是同一个项。

下面假设, 源知识域中的代数规范是完全的。可验证  $sp_{i:i}$  是完全的。

设  $sp_0 = (\Sigma_0, E_0)$  是用户给出的欲求解问题的代数规范,  $sp = (\Sigma, E)$  是源知识域中的代数规范, 验证  $sp \sim sp_0$  的过程可以简单描述如下:

(1) 根据化简关系把  $E \cup E_0$  中的等式定向成为重写规则得到重写系统  $R_0$  (例如我们把(1)式定向为  $\text{append}(\text{nil}, 1) \rightarrow 1$ , 因为规则右边的项是左边项的化简)。

(2) 设  $R_i$  是重写系统 ( $i$  的初值为 0), 求  $R_i$  的临界对的集合  $cp(R_i)$ 。所谓  $\langle t, s \rangle \in cp(R_i)$  是指存在  $l_1 \rightarrow r_1 \in R_i, l_2 \rightarrow r_2 \in R_i$ , 使  $l_1$  的位于  $\alpha$  的非变元子项  $l_1/\alpha$  与  $l_2$  之间存在  $\text{mgn}\sigma$  使得  $t \equiv \sigma(r_1), s \equiv \sigma(l_2) [\alpha \leftarrow \sigma(r_2)]$ , 即  $s$  是用  $\sigma(r_2)$  代换  $\sigma(l_2)$  中位于  $\alpha$  的子项所得到的项。例如,  $\langle \text{rev}(\text{append}(\text{rev}(1), a \cdot \text{nil})), a \cdot 1 \rangle$  是  $\text{rev}(\text{rev}(1)) \rightarrow 1$

和  $\text{rev}(x \cdot 1) \rightarrow \text{append}(\text{rev}(1), x \cdot \text{nil})$

的临界对。

a) 如果  $cp(R_i)$  为空, 则  $sp \sim sp_0$ , 过程终止。

b) 否则设  $\langle t, s \rangle \in cp(R_i)$ ;

i) 如果  $t, s$  是由  $C$  生成的不同项, 则  $sp$  与  $sp_0$  不相似, 过程终止。

ii) 否则, 如果  $s$  是  $t$  的化简, 则转 3。

(3)  $R_{i+1} = R_i \cup \{t \rightarrow s\}, i \leftarrow i + 1$ , 转 2。

具体的算法和证明请参见文 [11]。一些实验系统表明, 上面介绍的相似性准则和验证方法可作为一种软件重用技术。

## 4 结 论

本文中所提出的一类相似性准则及其验证方法是针对软件重用技术研究的。基于前面所阐述的思想, 我们研究设计了一个信息检索系统——KDZ<sup>[13]</sup>, 该系统是支持软件重用的。在 KDZ 系统中, 我们把软构件组成一个开放型的, 具有格结构的软件库, 把相似性准则作为匹配软构件的重用理论, 通过语法、语义项重写子系统来支持可重用验证。该系统改善了以往的软构件检索系统中缺乏语义检索的状况, 尽管在运行效率方面还有待于提高, 但本文所提出的原则和思想对软件重用技术的研究无疑是有意义的。

通过具体的实践, 可以对相似性准则作更具体的一般性规定:

(1) 相似性准则是一个与问题域及源知识域紧密相关的问题。

(2) 只有把问题域和源知识域置于一致的语义论域之中, 才能精确地规定相似性含义, 定义相似性准则, 进行相似验证。做到这一点的最理想的办法是用具有良好语义性质的语言, 描述欲求解问题和源知识。

(3) 相似是论域中的元素之间的某种逻辑相关性。相似性准则是这种相关性的刻划。

## 参 考 文 献

- [1] Carbonell J G. "Learning by Analogy" Formulating and Generalizing plans from Past Experience. Machine Learning, 1983, 1

- [2] Greifner R, Learning by Understanding Analogies, Artificial Intelligence, 1988, 35(1)
- [3] McDermott J. Learning to Use Analogy. IJCAI, 1979
- [4] Carbonell J G. A Computational Model of Problem Solving by Analogy. IJCAI, 1981
- [5] Tversky A. Features of Similarity. Psychological Review, 1977, 84
- [6] Goguen J A, Thatcher J W, Wagner E W. An Initial Algebra Approach to the Specification, Correctness and Implementation of ADT. CTPM IV, 1978
- [7] Afshar S K. Software Reuse Via Source Transformation. Proc. COMPSAC'85 Computer and Application Conf.
- [8] Biggerstaff T & Richter C. Reusability Framework Assessment, and Direction. Proc. of the Twentieth Annual Hawaii International Conf. on System Science, 1987
- [9] Luqi & Berzins, V. Rapid prototyping of Real-Time System. IEEE Software, 1988
- [10] Gaudel M and Moineau T. A Theory of Software Reusability, LNCS 300
- [11] Heut G & Hullot J. Proof by Induction In Equational-theories with Constructors. Jcss, 1982
- [12] 陈火旺, 王怀民. 重写技术及其在计算机科学中的应用. 计算机科学, 1990, (3)
- [13] 张少平, 王怀民, 陈火旺. 一个支持软件重用的信息检索系统—K DZ. 软件学报, 1990, (2)

## A Similarity Criterion and its Verification Method in Analogical Reasoning

Zhang Shaoping Wang Huaimin Chen Huowang  
(Department of Computer Science)

### Abstract

In artificial intelligent, analogical reasoning is an important field of study, and similarity criterion is a significant concept of analogical reasoning. The paper gives a descriptive framework of analogical reasoning, and a definition of similarity criterion. Based on term rewriting technique, the paper proposes a verification method of the similarity criterion.

**Key words** artificial intelligence, reasoning, specification/analogical reasoning, similarity criterion, term rewriting, software reuse