

## 重用部件的特征匹配

李留英 齐治昌

(国防科技大学计算机系 长沙 410073)

TP311.11

摘要 特征匹配是形式化规范匹配的基础。本文给出了基于运算符和抽象数据类型的特征匹配的定义、应用和展望。

关键词 运算符 抽象数据类型 精确匹配 松散匹配

重用部件, 软件重用, 特征匹配

在软件重用中,人们已提出各种方法来支持重用部件的检索,其中包括:浏览、关键字查询、多属性查询等。为了支持目标模型系统的开发,使原型尽量使用已有的部件,软部件库不仅包括部件的实现体,还包括部件的行为规范,行为规范描述了部件的特征和功能。系统根据用户需求规范,对软部件库的部件进行规范匹配以检索出所需部件。规范匹配包括特征匹配和语义匹配。所以,特征匹配是规范匹配(specification match)的基础。

本文介绍了特征匹配的基本思想,并采用 PSDL 语言描述软部件。PSDL 语言是规范、设计级的可执行原型描述语言,能刻画系统功能和模块功能,易于部件的检索。它提供两种抽象机制:运算符和抽象数据类型。PSDL 描述的软件系统是一个网络结构,结点表示运算符,结点间通过数据流相互通讯。设重用部件库中的存储部件为 SQC,用户要检索的部件为 SBC。下面给出抽象数据类型 Index 和 Bubble\_Sort 的规范说明。

```

TYPE Index
SPECIFICATION
    index, Range;
END
OPERATOR Bubble_Sort
SPECIFICATION
INPUT The_In_Array, ARRAY [ARRAY_EBEMENT,
    CHARACTER, ARRAY_INDEX, Index;]
OUTPUT The_Out_Array, ARRAY [ARRAY_EBE-
    MENT, CHARACTER, ARRAY_INDEX, Index;]
END
  
```

## 一、基本概念

给出软部件的形式化规范说明后,既可根据运算符的规范说明中的输入/输出参数表(包括异常信号)——过程类型(signature(Bubble\_sort)={ARRAY, [ARRAY\_EBEMENT; CHARACTER, ARRAY\_INDEX, Index;], ARRAY [ARRAY\_EBE- MENT; CHARACTER, ARRAY\_INDEX, Ind-

ex;])检索部件库,查询所需的运算符,也可根据抽象数据类型 ADT 规范说明中的接口描述——用户定义的类型和过程类型(signature(Index)={Range})检索部件库,查询所需的 ADT。例如,抽象数据类型 SET 有一个用户自定义类型 T 和几个运算符 OP,系统根据 SQC 的 T 和 OP 的集合,查找对应的 SBC,所以特征匹配过程就是 SQC 与软部件库内的部件匹配的过程。下面给出特征匹配的一般定义形式:

定义1 (查询特征,匹配谓词,部件库内部件) →部件的集合 Signature\_Match(q, M, C) = {c ∈ C, 且 M(c, q)}

其中, q 为给定的查询,描述所需部件的功能和使用方法,即所需部件的特征; M 为匹配谓词; C 为部件库。特征匹配返回一个部件集合,集合内的每个部件均满足匹配谓词。本文考虑两种类型的匹配,即过程类型匹配和抽象数据类型匹配,同时也考虑不同类型的匹配谓词:精确匹配和松散匹配。下面分别介绍运算符匹配和抽象数据类型匹配。

## 二、运算符匹配

运算符匹配以输入/输出参数类型为基础。参数类型为系统固有类型或用户自定义类型。每种类型均有相应的类型运算符,如“+”是二元运算符,我们用“=”表示类型的相等, typeop 表示类型运算符的集合。对类型 T 和 T', T=T' 当且仅当 T = typeop (t<sub>1</sub> ... t<sub>n</sub>), T' = typeop (t'<sub>1</sub> ... t'<sub>n</sub>), typeop = typeop',  $\forall i, 1 \leq i \leq n, t_i = t'_i$ 。对于相似类型,可通过变量替换实现类型变换。在软件设计过程中,即使对同一类型,不同用户可能会采用不同的名字,故可把用户自定义类型当作变量进行类型变换。根据上述理论,给出运算符匹配的类属形式:

$$M(t_b, t_q) = T_b(t_b) R T_q(t_q)$$

$t_b$  是查询部件的类型,  $t_q$  是部件库内部件的类型,  $M(t_b, t_q)$  表示类属匹配,  $T_b$  和  $T_q$  表示变换,  $R$  表示类型间的关系。一般只允许对其中的一个类型进行替换,  $R$  只起识别作用。

为简洁起见, 我们将松散匹配分为偏序匹配和变换匹配。偏序匹配中的  $R$  表示  $t_b, t_q$  间的关系, 变换匹配中的  $R$  代表对其中参数类型的替换。

### 2.1 精确匹配(Exact Match)

定义2 Match-Exact( $t_b, t_q$ ) =  $\exists$  改名序列  $V$ , 使  $V(t_b) = t_q$ 。

①对单一类型, 即运算符中只包含单个数据类型, Match-Exact( $t_b, t_q$ ) = ( $t_b = t_q$ ), 只需改变其中一个表达式的名字即可实现精确匹配。

②对多态类型, 即运算符中只包含多个同一数据类型, 只需同时替换某个类型的名称, 就可能进行匹配。如,  $t_b = (\alpha, \alpha) \rightarrow \text{bool}$ ,  $t_q = (\beta, \beta)$  采用变换  $V = [\alpha/\beta]$  就能实现匹配。

### 2.2 偏序松散匹配

由定义2可知, 精确匹配是一种较好的方法, 但也可能会遗漏有用的运算符, 这些运算符与查询类型比较接近又无法满足精确匹配的要求。当用户需要一般化(具体)类型的部件, 而部件库中只有该类型的某些具体化(一般化)类型的部件时, 两种情况均无法满足精确匹配的要求。为此, 定义通用匹配和特殊匹配以解决上述问题。

对精确匹配中的类属匹配,  $R$  表示类型间的相等关系; 对偏序匹配,  $R$  表示类型间的偏序关系。根据类型的通用性原则, 可以用变量替换来定义偏序排序。若  $\alpha \rightarrow \alpha'$  表示类型的一个通用形式, 不同的替换将产生不同的类型, 所以  $T$  比  $T'$  更通用些即  $T \geq T'$ , 当且仅当  $T'$  是用一系列的变量替换  $T$  的结果, 那么称  $T'$  是  $T$  的一个实例。

定义3(通用匹配) Match-Gen( $t_b, t_q$ ) =  $t_b \geq t_q$

如果部件库内的部件类型(简称库类型)比用户类型更通用些, 那么库类型与用户查询类型相匹配。精确匹配只是通用匹配的一个特例。所以对通用匹配实例化(用  $\Rightarrow$  表示)后就形成精确匹配, 即 Match-Exact( $t_b, t_q$ )  $\Rightarrow$  Match-Gen( $t_b, t_q$ )。

定义4(特殊匹配) Match-Spec( $t_b, t_q$ ) =  $t_b \leq t_q$

即用户查询类型比库类型更通用些, 也可根据特殊匹配定义通用匹配, 即 Match-Spec( $t_b, t_q$ ) = Match-Gen( $t_q, t_b$ ); 精确匹配也是特殊匹配的一个特殊实例化, Match-Exact( $t_b, t_q$ )  $\Rightarrow$  Match-Spec( $t_b,$

$t_q$ )。

由于类型间关系是根据变量替换来定义的, 所以可将通用匹配和特殊匹配定义为变换匹配, 形式如下:

Match-Gen( $t_b, t_q$ ) = 存在一个变量替换序列  $U$ , Match-Exact( $U(t_b), t_q$ )

Match-Gen( $t_b, t_q$ ) = 存在一个变量替换序列  $U$ , Match-Exact( $t_b, U(t_q)$ )

定义5(统一匹配) Match-Unify( $t_b, t_q$ ) = 存在一个变量替换序列  $U$ , Match-Exact( $U(t_b), U(t_q)$ )

### 2.3 变换松散匹配

这种匹配方法一般通过改换某个类型表达式的次序来获得。通常用多元对偶( $a_1, \dots, a_n$ )将多个变元组成一组, 变元的次序无关紧要。如测试集合内成员的运算符  $OP$  具有类型( $\alpha, \alpha, \text{list}$ )或( $\alpha, \text{list}, \alpha$ )。由此定义了重排匹配。

定义6(重排匹配) 当库类型与用户查询运算符类型匹配时, 重新排列库内部件参数类型可获得精确匹配。

### 2.4 松散匹配的合成

每种松散匹配均是较多的搜索方法, 它们的联合将会扩大检索的范围, 由2.2、2.3等联合起来给出松散匹配的一般定义。

存在一个变换对偶  $T = (T_1, T_2)$ , 使 Match-Exact( $T_1(t_b), T_2(t_q)$ )。变换对偶( $T_1, T_2$ )是一个变换三元对偶, 描述库类型和查询类型的变换。两个松散匹配的合成表示为: Match-T1. Match-T2, 定义如下:

定义7(匹配合成) (Match-T1. Match-T2)( $t_b, t_q$ ) = 存在变换对偶( $T_1, T_2$ ), Match-Exact( $T_1(T_2(t_b), T_1(t_q), T_2(t_q))$ )。

用户就能以任意次序组合松散匹配, 完成查找任务。

## 三、抽象数据类型匹配

绝大多数高级语言均支持抽象数据类型, 如 ADA 程序包, C++ 类等, 它将描述自然对象的数据和对数据的操作封装在一个抽象数据。运算符的特征是指过程的类型, 抽象数据的特征是指它的接口: ( $L-T, L-P, L-E$ )。L-T 表示封装在抽象数据内的数据类型的集合, L-P 是过程类型的集合, L-E 表示异常信号及相应的处理集合。若部件库内部件的接口  $L-B = \langle L-BT, L-BP, L-BE \rangle$  与查询接口  $L-Q = \langle L-$

QT, L-QP, L-QE) 匹配, 则 L-BT 与 L-QT, L-BP 与 L-QP, L-BE 与 L-QE 之间必须有对应关系。抽象数据类型特征匹配也包括精确匹配和松散匹配。

### 3.1 精确匹配

定义 8 C-Match-Exact(L-Q, L-B) = 存在一个映射  $Uf: L-QT \rightarrow L-BT, L-QP \rightarrow L-BP, L-QE \rightarrow L-BE$ ,

$Uf$  是一个内射函数,  $\forall t_q \in L-QT, Match-Exact(Uf(t_q), t_q), \forall p_q \in L-QP, Match-Exact(Uf(p_q), p_q), \forall e_q \in L-QE, Match-Exact(Uf(e_q), e_q)$ 。由于是一一对应关系, 所以两种接口的运算符个数必须相等, 且查询运算符类型能映射到库类型。

就大多数抽象数据类型来说, 运算符匹配也包括用户自定义数据类型的匹配, 所以一个接口中出现的用户自定义类型必须与另一接口内用户定义数据类型相匹配, 即结构等价或定义等价。

### 3.2 部分匹配

部件检索时, 有时用户只需列出抽象数据内感兴趣的运算符, 就可以检索出更通用的部件。被检索出的部件包含了查询者感兴趣的所有运算符和其它运算符。为此, 定义了通用匹配和特殊匹配, 含义与前面类似。

定义 9(通用匹配) C-Match-Gen(L-Q, L-B) = 存在一个映射  $Uf: L-QT \rightarrow L-BT, L-QP \rightarrow L-BP, L-QE \rightarrow L-BE$ 。

$Uf$  是一一对应关系,  $\forall t_q \in L-QT, Match-Gen(Uf(t_q), t_q), \forall p_q \in L-QP, Match-Gen(Uf(p_q), p_q), \forall e_q \in L-QE, Match-Gen(Uf(e_q), e_q)$ 。在 C-Match-Gen(L-Q, L-B) 中,  $|L-QT| \leq |L-BT|, L-QP \subseteq L-BP, L-QE \subseteq L-BE$ 。

定义 10(特殊匹配) C-Match-Spec(L-Q, L-B) = C-Match-Gen(L-B, L-Q)。

在特殊匹配中, 库部件不必包括查询部件中的所有运算符。但两部件关于异常的处理动作必须相同。

### 3.3 松散匹配(Relax\* Match)

定义精确匹配时, 我们用精确匹配谓词判断查询接口内的运算符是否与部件库的部件接口内的运算符匹配, 在松散匹配中, 用松散匹配参数代替精确匹配谓词。

定义 11(松散匹配) C-Match-Relax\*(L-Q, L-B, Mf) = 存在一个映射函数  $Uf: L-QT \rightarrow L-BT, L-$

$QP \rightarrow L-BP, L-QE \rightarrow L-BE, \forall t_q \in L-QT, Mf(Uf(t_q), t_q), \forall p_q \in L-QP, Mf(Uf(p_q), p_q), \forall e_q \in L-QE, Mf(Uf(e_q), e_q)$ 。

松散匹配与精确匹配唯一的差别是松散匹配用参数代替谓词。匹配参数提供了很大的灵活性, 允许使用第二部分定义的各种匹配方法来匹配抽象数据类型接口内的单个运算符。对异常信号, 可通过改名或扩充方法来辅助匹配。

### 3.4 松散匹配的合成

与运算符类似, 我们也可以定义抽象数据类型匹配的合成。既然由通用匹配可以定义特殊匹配, 那么只需考虑通用匹配和松散匹配的合成, 且合成次序无关紧要。

定义 12(通用松散匹配) 除  $Uf$  不必是从定义域到值域上的映射外, C-Match-Gen-Relax\*(L-Q, L-B, Mf) 与 C-Match-Relax\*(L-Q, L-B, Mf) 基本相同。

将这个定义单独列出来, 是希望这个联合松散匹配将成为最通用的匹配方法。

小结和展望 本文介绍了运算符匹配和抽象数据类型匹配的各种定义。联合松散匹配仍然会漏掉一些有用的信息, 为此, 有必要扩充现有的类型系统, 增加对实时、中断、并发的处理。

特征匹配可视为传统信息检索技术的补充, 例如特征匹配和分类方法结合形成“管道”查询, 先采用分类机制, 再使用特征匹配, 也可将特征匹配扩充为规范匹配。软部件库中的领域信息, 不仅包括部件的特征, 还包括对每个部件行为的形式化描述。给定这样的规范, 如过程的前置条件和后置条件, 运算符的功能描述—公理, 加上适当的推理机制, 就可构成规范匹配器。规范匹配将部件的行为和语法区别开来, 并将特征和规范作为搜索关键字。

特征匹配为面向对象或具有面向对象特点的系统提供了最大限度的信息和最小的负载, 我们可以充分利用它的特性来开发大型软件系统。

### 参考文献

- [1] Amy Moormann Zaremski and Jeannette M. Wing, Signature Matching, A Key to Reuse
- [2] Luqi, Valdis Berzins, et al., A Prototyping Language for Real-Time Software, IEEE Tran. on Software Engineering, Vol. 14, No. 10