形式概念分析在软件工程中的应用

杨帆

作业	分数
得分	

2020 年 11 月 13 日

形式概念分析在软件工程中的应用

杨帆

（大连海事大学 计算机技术 辽宁省大连市 中国 116026）

摘 要 随着计算机技术的发展,对于软件开发效率提出了更高的要求，传统软件开发过分地依赖于文档，开发效率不高且灵活性不强。形式概念分析(Formal Concept Analysis, FCA)理论通过对数据集中对象和属性之间的二元关系建立概念层次结构，生动简洁地体现了概念之间的泛化和特化关系，再运用格代数理论对数据进行分析。为此，将 FCA 与概念格技术引入软件工程领域，根据软件开发的不同阶段，介绍 FCA 与概念格技术在该环节的具体应用方式及优缺点，并在此基础上分析出在软件完整性、合格检测及大规模软件开发应用中等进一步需要研究的内容。

关键词 形式概念分析；概念格；软件工程

中图法分类号 TP311.20 **DOI 号** 10.3969/j.issn.1001-3695.2014.01.030

Application of FCA in software engineering

Yang Fan

Department of computer, Dalian Maritime University, City Dalian

Abstract With the development of computer technology, higher requirements are put forward for the efficiency of software development. Traditional software development relies too much on documents, and the development efficiency is not high and flexibility is not strong. Formal Concept Analysis (FCA) theory establishes a conceptual hierarchical structure of the binary relationship between objects and attributes in the data set, which vividly and concisely reflects the generalization and specialization relationship between concepts, and then uses lattice algebra theory Analyze the data. To this end, FCA and concept lattice technology are introduced into the field of software engineering. According to the different stages of software development, the specific application methods, advantages and disadvantages of FCA and concept lattice technology in this link are introduced, and on this basis, the software integrity, Qualified testing and large-scale software development and application, etc. need further research.

Key words Formal Concept Analysis; Concept Lattice; Software engineering

0 引言

随着如今科学技术的进步和人们对于创新和高科技成果的使用，软件工程已经成为了工作和生活当中的热门方向，在实际当中也有许多的工作者会投身到软件的开发和管理当中。所以在这个现代软件工程不断进步和高速发展的时代当中，软件开发的质量和效率也引起了越来越多人们的关注。而且在这个创新理念和探索精神被高度提倡的年代当中，怎样才能够高效的开发出满足不同层次的客户需求也已经成为了现代软件开发的重要热点。相比于现今的软件开发和社会需求而言，传统软件开发对于文档有着过分的依赖性，不利于灵活有效的开发软件。形式概念分析方法已经逐渐融入到软件工程这一学科，对象和属性之间所建立的二元关系经常出现在软件相关领域 [1]，对这种关系的处理方法也间接地推动了形式概念分析在软件测试等其它方面的应用。

1 软件工程概述

从上世纪 80 年代开始大多数研究机构进行了软件工具和软件开发环境的研究，提高了软件开发的效率，支持软件生存期模型的软件开发环境也进入了实用化阶段。软件开发环境软件工具是软件工程发展的重要基础，也是关键性、综合性研究课题。

根据传统的软件生命周期模型来看，软件生命周期由软件定义、软件开发和软件维护（也称为运行维护）三个时期组成，每一个时期又进一步划分成若干个阶段，如图 1 所示。

软件定义时期又可以划分为软件计划、需求分析、软件设计三个阶段。这一时期主要是确定要做的软件“是什么”，对软件进行顶层设计，描绘出软件架构，并对目标软件系统提出完整、准确、清晰和具体的要求。软件开发时期主要包括程序编码和软件测试两个阶段．根据实际情况选择合适的编程语言，并编写测试计划、测试分析报告等文档，用合理的测试方法（黑盒、白盒等）来逐步测试并改正系统中的错误。软件维护是软件生命周期的最后一个阶段，也是持续周期最长、花费代价最高的一个阶段。虽然是软件投入运行之后需要进行的工作，但软件维护的工作通常会占用软件开发机构60%以上的精力。[2]

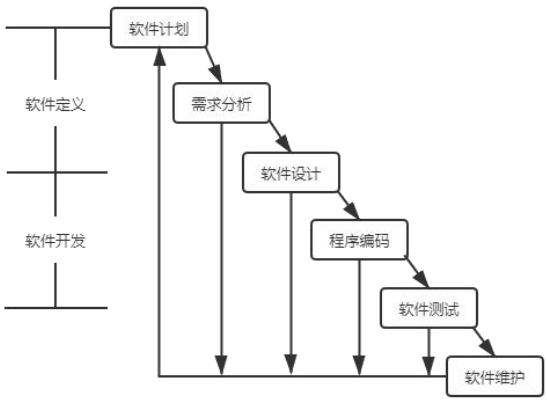

图 1 传统软件工程生命周期

2 形式概念分析

形式概念分析是建立在数学理论与概念理论之上的，可以说它是应用数学当中的一个重要分支。可以说它是在较大范围当中来搜集对于数学概念的“集合”有重大帮助的一个元素，而且还需要使用有关的形式概念分析方法来实现、构造和展示对象与属性之间的有机关系。正是因为形式概念这种独有的属性，这种方法也已经被适当的应用到了软件开发的很多环节当中。此外，形式概念这种分析方法在研究过程当中的工具主要是线路图，而且这种线路图本身就是对于概念格的图形化的表示。线路图主要是包含着在语境当中的对象和属性这两者之间的关系，这也可以说是语境当中另外一种表现形式。而在特定语境当中也是包括继承与发展的，所以说形式概念分析是一种较为准确和使用率较高的方式。

定义 1[3] 称 T:=(G,M,I)为一个形式背景，其中，G 是一个对象集，M 是一个属性集，I 是 G 和 M 之间的一个关系。分别称 G 和 M 的元素为对象和属性。

若对象 g 和属性 m 具有关系 I，则记为$(g,m)\in I$或 gIm 。

定义 2[3] 设 T=(G,M,I) 为形式背景， $X\subseteq G$，$B\subseteq M$。若二元组 (X,B) 满足 X'=B，B'=X，则称 (X,B) 为形式概念，其中：

$$X'=\{m\in M\mid \forall g\in X,(g,m)\in I\}$$

$$B'=\{g\in G\mid \forall m\in M,(g,m)\in I\}$$

设 T=(G,M,I) 是形式背景，对任意的形式概念 (X1,B1),(X2,B2) 定义如下偏序关系：

$$(X_1,B_1)\le(X_2,B_2)\Leftrightarrow X_1\subseteq X_2(\Leftrightarrow B_1\supseteq B_2)$$

记 L(G,M,I)或 L(T)为 T=(G,M,I) 中所有形式概念构成的集合，则 L(T) 是格，称其为 T 的概念格。在概念格 L(T)上定义上、下确界如下：

$$(X_1,B_1)\wedge(X_2,B_2)=(X_1IX_2,(B_1\bigcup B_2)'')$$

$$(X_1,B_1)\vee(X_2,B_2)=((X_1\bigcup X_2)'',B_1IB_2)$$

则概念格 $(L(T),\wedge,\vee)$ 是一个完备格。

性质 1[4] 设 T=(G,M,I) 是一个二元背景，X,X1,X2 为任意对象子集，B,B1,B2 为任意属性子集，则下列性质成立：

(1)若 $X_1\subseteq X_2$，则 $X_1'\subseteq X_2'$；

若 $B_1\subseteq B_2$，则 $B_1'\subseteq B_2'$

(2) $X\subseteq X'',B\subseteq B''$

(3) $X'=X'''$， $B'=B'''$

$(4)(X_1 \cap X_2)' \supseteq X_1' \cup X_2', (B_1 \cap B_2)' \supseteq B_1' \cup B_2'$

$(5)(X_1 \cup X_2)' = X_1' \cap X_2', (B_1 \cup B_2)' = B_1' \cap B_2'$

$(6)(X'', X') \in L(T)$

$(7)X \subseteq B' \Leftrightarrow B \subseteq X'$

3 软件工程中形式概念分析法的具体应用

3.1 形式概念分析方法在需求分析中的应用

形式概念这种分析方法是在近年来才兴起的，在这种分析方法出现之后就被广泛性的使用到了各个领域当中。而需求分析则是软件在不同需求当中应用性的分析，这也就表明在分析的过程中能够具体分析其所使用的各种环境，之后则能够对不同类别软件的应用信息进行有效的整合与管理。所以从大的方向来看，形式概念这种分析方法能够促进在各种实际应用当中软件使用的灵活性和有效性。具体到其实际使用的要求来看，这种分析方法的语境和属性都能够在需求分析当中起到非常重要的作用，因而也就能够为相关和同类别的软件提供一些基础和借鉴。[5]

应用在软件工程需求分析阶段主要是构造软件项目特征集。这个过程就是语境及属性集的构造。如表 1 所示。

表 1 对象--属性集

属性 / 对象	需求分析	结构设计	详细设计	编码	测试	综合	资格测试	安装	认同支持	软件维护
调试					×					×
概念					×					×
用例			×							×
识别	×	×								
再工程			×							×
控制	×									
帮助	×									
环境			×							×

表 1 中:“×”表示连接对象和属性的符号，比如说 (调试，测试) 可以组成一个对象--属性集, 表示了其在软件工程中操作的优先级别和对象属性。同时，对于其他部分的开发工作，也可以通过对其具体的特征集合进行收集。

通过对软件工程中各个传统模式下所对应的部分工作集和其属性的对应关系， 运用概念分析的方法可以得到表 1 所示的项目特征集合，从而也为软件的需求分析奠定了基础 。

3.2 形式概念分析方法在结构设计中的应用

软件设计和研发作为一种高科技领域范畴的工作，主要是对于数据结构进行改革和提升。所以在基本需求分析和管理的基础之上对其数据进行合理化的组织和分析，这样才能够总结出较为科学和有效的方法。一般来说假设这种方式能够较为合理的表明概念分析方法当中对于概念的构造原理，之后则可以通过有关的分析器来对每个项目进行分析和使用。因而在形式概念分析的过程当中需要具体分析变量之间的关系，这样才能够分析不同项目特征所能够形成的相关概念，才能够形成系统化的概念格，也才能够更加有效的促进形式概念这种分析方法在结构设计中的有效使用。

3.3 形式概念分析方法在系统设计中的应用

在软件工程系统设计当中，这一阶段最主要的任务和工作就是需要构造出系统的概念格，之后则能够在此基础上出现较多概念格的构造算法。而这种算法是能够更加科学有效的计算出格的概念和不同层次的关系，这样才能够具体分析出不同项目所组成集合的核心内容。在寻找到概念分析法的核心之后就可以较为准确的获得相关概念间的关系。通过对于不同集合关系进行分析和整合就可以获得系统化的概念格。在之后相关专家就可以通过一系列的计算来对概念格的数值范围进行分析和计算，这样才能够促进软件应用的最大化。

3.4 形式概念分析方法在软件测试中的应用

IEEE 中提到软件测试的目的在于检验它是否满足规定的需求或者弄清预期结果和实际结果之间的差别．概念格能够反映面向对象程序的特征，体现面向对象程序的类层次，可用于对程序结构的了解和软件分析。

Khor 和 Grogono 提出基于遗传算法和 FCA 自动产生分支覆盖的测试数据[6]．他们将这种方法命名为”genet”，通过使用遗传算法寻找测试，然后使用 FCA 技术组织测试和测试的执行跟踪之间的关系。回归测试是软件测试的重要方法，国内学者易利通过介绍概念格如何反映面向对象程序的特征，用概念格体现面向对象程序的类层次。针对面向对象程序的特征，利用概念格的层次聚类特性，将类和类的成员方法作为对象，类成员方法作为属

性构造概念格，得到面向对象程序的类层次结构。当类成员方法修改时，通过分析各个概念之间的偏序关系，找出可能受到影响的类成员方法和类。

3.5 形式概念分析方法在软件维护中的应用

软件维护包括矫正、适应性、完善、预防等方面，FCA 在软件维护方面的应用覆盖了上面所提到的四个方面，而且这些应用有一个共同点，即通过组织软件系统的构件，提取可以理解的结构。这些应用的不同则取决于输入、构造的概念格以及将概念格应用在哪里。为了组织和呈现出这些不同的方法，Thomas 等人将它们分为如下几类：动态信息分析，在遗留系统方面的应用和类层次的检查。

动态信息分析：Dekel 使用概念格技术对 JAVA 类中的代码进行检查，根据类中成员作用的领域，构造概念格，然后通过将“方法调用图”附加到之前的类层次概念格上，得到一个 “嵌入式调用图”，它可以提供更多关于类的详细信息，从而实现类结构的可视化，更易于对单个类的理解。下面给出一个关于类 Point3D 的例子，对这个方法介绍如下：首先，用一个自动的程序从类文件中提取方法和访问区域的信息，使用标准静态分析技术，将这些信息用来计算每个方法的访问区域，包含非直达区域。根据概念格的定义，得出类 Point3D 的形式背景如图 2 所示。从以上可以看出，在软件维护阶段，对于不可用的源文件，使用概念格技术与“嵌入式调用图”更有利于实现类的重构。由 Ball 提出结合概念格使用动态分析来分析正在运行程序的性能．动态分析有效性取决于它的两个特征：信息的精确性和对输入的依赖。使用测试集和指令实体建立多值背景，进而转化为单值背景，建立概念格。

遗留系统方面应用：Poshyvanyk 等借助形式概念分析，结合潜在语义索引对开发者在识别源代码的过程中进行概念定位[7]。这种基于概念定位技术的信息检索，是使用搜索引擎并结合潜在语义索引，允许使用者搜寻源代码及与之相关的文件，并根据搜索到的元素自动建立概念格。

类层次的检查：由于很多软件在设计的时候不能很好地遵循面向对象的特点和设计原则，使得软件缺乏模块化，从而造成软件维护困难，不易对软件进行功能扩展。另外， Hakik 等人基于形式概念分析将软件体系结构重新模块化，通过测量耦合和内聚来估计被重新模块化后的系统性能。Cellier 则将形式概念分析用到故障跟踪方面。排除故障是软件开发过程中比较费时的一项工作，而将形式概念分析与关联规则相结合，就可以用来分析执行代码中的错误，提高开发效率。

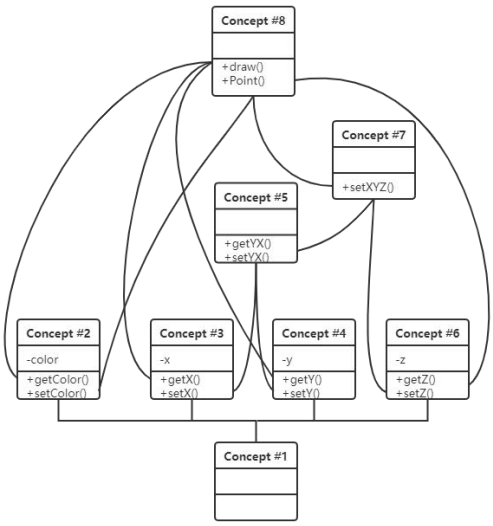

图 2 类 Point3D 嵌入式调用图

4 结束语

在软件工程领域，形式概念分析与概念格还被应用于组件重构、设计帮助等诸多方面。这些应用使得在软件开发过程中，客观成分逐渐加大，形式化的特征逐步增强，标准化趋势更加明显。另一方面，这些应用使得软件工程对个体认知的依赖程度逐步减弱，受研发人员个体素质的影响逐渐消失，进而大大提高了软件设计、开发、维护、重用的效率和效益。目前对概念格的应用还是一些概念分析中初步应用，并没有涉及到概念格中更丰富的理论和代数应用；其次，在软件完整性、合格检测、接受支持或编码方面的应用较少，这些方面将是 FCA 要研究的领域。

参 考 文 献

[1] 孙小兵, 李云, 李必信,等. 形式概念分析在软件维护中的应用综述[J]. 电子学报, 2015, 43(007):1399-1406

[2] 臧国轻, 李瑞光, 郑珂. 形式概念分析在软件工程中的应用综述[J]. 河南大学学报(自然科学版), 2018, v.48(03):60-68

[3] Wille R , Fachbereich T H D . Conceptual Graphs and Formal Concept Analysis[J]. Lecture Notes in Computer Science, 1997

[4] Wille R . RESTRUCTURING LATTICE THEORY: AN APPROACH BASED ON HIERARCHIES OF CONCEPTS[C]// International Conference on Formal Concept Analysis. Springer, Berlin, Heidelberg, 2009

[5] 胡鑫. 形式概念分析在软件工程中的应用[J]. 电脑迷, 2016, 000(005):39

[6] Khor S , Grogono P . Using a genetic algorithm and formal concept analysis to generate branch coverage test data automatically[C]//

[7] Poshyvanyk D , Marcus A . Combining Formal Concept Analysis with Information Retrieval for Concept Location in Source Code[C]// Program Comprehension, 2007. ICPC '07. 15th IEEE International Conference on. IEEE, 2007