

Ada语言

Ada

源代码分析器

源代码分析器

(23)

Ada 软件源代码分析器

89, 封回, 65

钱红兵 徐红

TP312Ad

(北京航空航天大学软件工程研究所 北京 100083)

摘要 This paper describes the structure, function and some features of ASCA. ASCA is one of the tools for ensuring the quality of Ada Software system. It can support the development and maintenance of great and complex software system effectively.

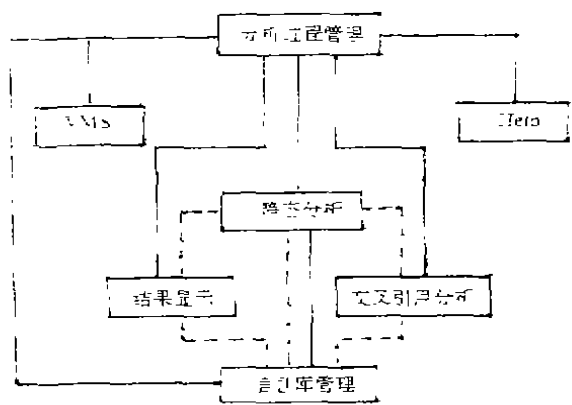
关键词 Static analysis, Information base, Across reference analysis.

Ada 语言是美国国防部(DOD)为解决软件危机和日益增长的软件开发费用而主持研制的一个通用性强,能支持现代程序设计方法学的高级语言,旨在满足大型、实时、嵌入式计算机系统的软件设计需要,它与其它通用的程序设计语言相比,增加了程序包、类属程序单元、任务、异常处理及子程序名和操作符重载等设施,可以实现信息隐藏、数据抽象、并发处理、异常处理和类属定义,增强了语言的能力,但因此也使语言变得十分复杂,给 Ada 软件的分析理解增加了难度。

Ada 软件源代码分析器就是针对 Ada 语言的特点而开发的,是对大型复杂的软件系统进行分析与理解的重要工具,也是 Ada 软件质量保证系统中的工具之一,它由分析过程管理、信息库管理、静态分析、分析结果显示及交叉引用分析组成,已在 VAX/VMS 系统上开发和运行。

一、系统结构

Ada 软件源代码分析器的总体结构和信息流如下图所示:



— 表示控制关系 - - - 表示信息流关系

二、分析过程管理及信息库管理

系统的分析过程管理通过选单驱动系统引导用户进行分析显示活动,并把它们集成为一个有机整体,以实现整个系统的功能。它与 VMS 之间设有接口,用户不退出系统就可执行 DCL 命令。它还设有 Help 功能为用户提供帮助。

信息库管理负责对信息库进行统一的管理,被分析软件的各种结果信息存放于信息库中,对信息库的管理包括:信息库的创建、初始化、删除以及删除信息库中文件等功能。它是建立在文件系统的基础之上的,每个信息库都是一个 VAX/VMS 子目录。通过库管理,用户可以建立多个信息库,每个信息库对应于一个软件系统(项目),当用户对一软件系统进行分析时,首先要建立一个项目文件,记录被分析的软件系统(项目)的所有文件名以及它们存放的路径名,然后通过信息库管理,创建该项目对应的信息库,并对该信息库进行初始化。初始化工作包括:建立项目文件与该信息库的对应关系(如果已有不用再建);调整指向项目文件中的指针,由该指针指定本次分析从项目文件中的第几个文件开始,其指针调整之前记录的是上次分析后未被分析过的第一个文件名;对项目文件中的文件进行格式化处理(由指针处开始);记录整个项目的分析历史;复制一些 Ada 标准程序单元(如 Text_IO 等)经过静态分析产生的信息以及恢复过去已经过各种分析产生的结果(如果有的话)等,凡使用本工具者首先要进行信息库的初始化。

三、静态分析

静态分析是本工具的基础。在不运行所分析文件的情况下,对通过 Ada 编译和格式化处理的 Ada 源程序(由项目文件指出)进行扫描和提取,生成被分析程序的静态属性信息,这些信息不仅可供质量

自动判断并执行之,直至完全平面化,我们将连续使用 μ 运算直至完全平面化的运算叫做平面化运算(Flatten),nest 运算也应连续执行,直至完全嵌套化,但此过程难以由系统自动完成,需人工干预,但第一次使用 nest,至少应针对一个对象标识进行,以使此对象标识消重,变成顶层对象标识,即嵌套化后形成的对象的对象标识(不再是纯元组),这里 Π^1 运算起重要作用。

例3 对例1,运算

Flatten ($\Pi_1^1(\text{dept})$), $Q = (\text{Dept. D-NO}, \text{Dept. D-name})$;

Flatten ($\Pi_2^1(\text{dept})$), $Q = (\text{Dept. D-project}, \text{P-NO}, \text{Dept. D-project}, \text{P-name})$;

Flatten($\Pi_3^1(\text{dept})$), $Q = (\text{Dept. D-Project}, \text{members. E-NO}, \text{Dept. D-project}, \text{members. E-name})$;

这三个运算得到三个平面关系,其第三个的结果示于图7,其它二个省略之。

D ₁	P ₁	E ₁	201	Davie
D ₁	P ₁	E ₂	202	Smith
D ₁	P ₂	E ₂	202	Smith
D ₁	P ₂	E ₃	203	John
D ₂	P ₃	E ₃	204	Jemes
D ₂	P ₃	E ₁	205	Anna

图7 Flatten 之例

此结果与文[3]得到的平面关系相同,后者不是通过代数运算得到,而只做了直观解释。

对 Π^1 运算结果平面化所得到的关系中含有标识

路径信息,便于和复杂对象代数中的路径表达式相对应,有利于相互转换,同时,标识路径中的标识实质上就是外关键字,保留了复杂对象所表达的实体间的联系,这种平面关系,相当于对应于联系实体的关系,并且,所得平面关系的规范化程度较高,这些都优于文[1]中的 μ 和文[2]中的 unnest,当然,平面化的方式不是唯一的。

参考文献

[1]Roth M. A. et al., Extended algebra and calculus for nested relational databases, ACM TODS, 13 (4), 1988
 [2]Shaw G. M. et al., An object-oriented query algebra, Proc. 2nd Int. Workshop on database programming language, 1989
 [3]Meng W. et al., Construction of a relational front-end for object-oriented database systems, IEEE Data Engineering Conference 1993
 [4]李天柱,对象标识的语义及构造,计算机科学,20(5),1993
 [5]李天柱,NF²关系模型与属性继承,软件学报,6(4),1995
 [6]Kim W., A model of queries for object-oriented databases, VLDB, 1989
 [7]Scholl M. H. et al., A relational object model, Proc. 3rd Int. Conf. on Database Theory, 1990
 [8]Kim W., Introduction to object-oriented databases, The TIM Press, 1990

(上接封4)

哪行,包含它的源程序代码等信息,另外用户也可以通过点取交叉引用关系中的单元名,将它的下一级交叉引用关系显现出来,显示的另一种方式是一次就把整个软件系统的交叉引用关系从根结点到叶子结点全部显示给用户,它包含调用的单元名以及 WITH 的过程名、包名,USE 的包名,同样,用户也可以通过鼠标的点取操作来显示上一种方式所显示的那些信息,另外,用户还可以用不同的颜色把交叉引用关系图中的全部过程名及其它它们之间的调用关系或各任务名及其它它们之间的通讯关系突出显示出

来,供用户对顺序执行的程序以及并行执行的程序进行分析、理解。

所产生的交叉引用表及未被使用过的单元名表(如果存在)也是以文件形式存于信息库中的。

ASCA 首先是在 VAX 机上开发的,现在已移植到 ALPHA 机上,有关的界面及图形将用 Motif 实现,我们力求为用户提供一个用户界面友好、操作简单方便、能真正帮助用户分析、理解大型 Ada 软件项目的工具,相信它会为提高 Ada 软件系统的开发效率、保证其可靠性发挥不小的作用。(参考文献共5篇略)