

现有 Z 的工具的比较*

Comparison of Current Tools for Z

庞军 王云峰 郑国梁

(南京大学计算机科学系 南京210093)

Abstract Based on the analysis and comparison of the six typical supporting tools for Z and the characteristics of our COOZ-Tools, shortcoming of all the tools is summarized and further research is proposed.

Keywords Tools for Z, Z specification, Proof, Refinement

一 引言

软件开发过程需要大量的工具支撑,如编辑工具,编译工具,调试工具等。这些工具的使用不仅方便了软件的开发,而且合适的工具保证了软件开发的质量,它将使软件开发人员的注意力集中到设计工作本身,而不耗费在设计的相关工作上。

形式化方法为计算机系统的规约、实现和验证提供了合理的数学基础,首先对系统进行独立于实现的、基于一定形式化语义的抽象描述,然后经过逐步求精及变换,利用规约的数学特性检验其一致性,直至生成最终的软件系统。其优点在于它严格、精确地定义了用户需求,形式化规约的求精过程具有可推导、易证明的特性,从而有利于保证程序的正确性和实现软件自动化。但由于形式化方法是建立在数学基础上,含有大量的图示符号,因而在计算机中的编辑和显示会产生困难,同时对形式化规约的求精和推理过程也缺少很好的工具的支持。因此对形式化方法缺乏合适的工具支持,这一点已经制约了形式化方法的具体应用。

目前 Z 已发展成为最为流行的形式化规约语言之一,在欧洲、美国的工业界以及其他一些地区的软硬件开发项目中都得到应用。提供一个良好的 Z 支撑环境显得格外重要。从九十年代起,国际上对 Z 支撑工具的研究日趋活跃,目前已出现十余种 Z 支撑工具的版本。本文对现有国际上主要的 Z 支撑工具的优缺点进行分析与比较,针对其不足之处指出在我们目前研究开发的 COOZ 的支撑环境 COOZ-Tools 中进一步的研究方向和工作。

对于每一种工具我们主要从工具支持的平台,编

辑和浏览规约,规约的精确化和推理,对大型规约的支持以及工具本身的开放性和对多用户的支持等几个方面来描述基本的功能,然后再对其主要优缺点加以分析,最后我们从以上所列的几个特性进行列表评价,综合比较。

二 现有 Z 的工具的比较

现有的 Z 支撑工具主要集中于规约的编辑和显示以及语法和类型检查方面,也有些工具支持规约的推理和精确化。较有影响的工具有 Formaliser、Proof-Power、Zola、CADiZ、ForMooZ、Z/EVES 等。

Formaliser 由 Logica Cambridge 在微机 Windows 环境下用 Smalltalk/V for Windows 系统实现的以图形方式支持 Z 规约语法制导编辑、浏览工具和类型检查工具。它采用语法制导的风格,在用户进行规约编辑的同时立即进行语法、类型检查。Formaliser 采用结构编辑和短语输入相结合的输入方式,以保证所编辑的 Z 规约的语法正确性。搜索功能仅局限于文本字符串。它有两种浏览方式:适合于阅读的符号浏览方式和用于打印的扩展 LATEX 格式文本显示方式。对 Z 规约的类型检查功能集成在编辑器中,但必须每一段落分别进行检查,缺少整体检查功能。它有一些很好的特点,如:支持用户对变量使用范围和表达式结果类型的交互式查询;用符号板进行 Z 符号的输入;可通过多个文档表示一个规约;提供相应的语法属性描述具有通用性,Formaliser 也可以支持其他形式语言;等等。但 Formaliser 也存在一些不足:语法制导的编辑方式对初学者提供很多帮助,但操作起来却较为繁琐;即时的语法检查虽然可有效地减少错误,但一次只能检查

* 本文的研究工作得到国家九五攻关项目以及国家自然科学基金(69673006)的资助

出一段中的一个错误;非形式化的说明信息只能出现在规约的最外层;缺少联机帮助功能等。Formaliser 不支持规约的精细化和推理。对大型规约的支持,采用类似于 C 语言中头文件的方式将引入已命名规约的方式来解决,Formaliser 被设计成单用户在 PC 机上使用,不支持多用户使用,但由于该工具的底层功能被设计成类似于单片集成电路的易于访问的结构,所以工具本身很容易被修改和扩充。

ProofPower HOL 是 ICL 开发的商用定理证明工具,ProofPower Z 是 ProofPower HOL 的扩充,是 Z 的定理证明和文档准备支撑工具。PowerProof 在 SUN 工作站的 Sunview 系统下运行。ProofPower 采用标准的 Sunview 文本编辑器编辑 Z 规约,ICL 为 HOL 和 Z 开发了专门的 Sunview 字体进行数学符号的输入和显示,具有对文本和符号的搜索替换功能。可将规约转换为 LATEX 源文件用于打印,也可将其转换为标准 ML 源文件以作为 ProofPower 定理证明工具的输入文件。类型检查包含在定理证明工具中,主要是针对 Z 的表达式进行类型检查,对 Z 的定理证明通过将 Z 公式转换为 HOL 公式进行,虽然一般来说并不需要使用者熟悉 HOL,但由于 Z 和 HOL 之间的差异,这种间接的证明方式可能存在着局限性。定理证明工具可用于求精过程中的验证,但工具中没有支持将规约精细化为一现存编程语言。ProofPower 支持中等规模的规约,但目前还不适合于对大型规约的输入编辑,主要是由于当规约的规模很大时,可执行图像的规模也大幅度提高。ICL 声称 ProofPower 中所用到的功能可以为用户再使用。

Zola 是 Z 规约的创建、维护和验证工具,它由剑桥皇家软件研究所用 Lucid Command Lisp for Suns 开发,运行环境是 SunOS4 的 Sunview 或 X Motif。为了定理证明的需要,它支持的是 Z 的一个扩展版本。Zola 提供两种规约输入方式:语法制导的编辑器和文本直接输入法,用户可根据对 Z 语法的熟悉程度选择使用。浏览器中规约以标准格式显示,也可进行隐藏细节的抽象显示,可生成 PostScript 格式的打印文件。对于过长的规约可分段处理,在编辑的同时可保证语法的正确性,类型检查可通过内置的类型检查程序分段进行。Zola 的定理证明器将目标分解为复杂度小的子目标进行处理,用户可自由决定证明工作到何时结束,它还提供了一致性检查过程中的自动辅助。Zola 中没有提供将规约精细化到代码的功能。用 Zola 来产生大型规约,但在类型检查和定理证明时将会由于目标规模巨大而难以操作。Zola 支持多用户使用,它提供的 Z 的工具箱可以为用户共享使用。但对 Zola 的扩展很难做到。

CADiZ 由 York 软件工程公司开发,用 C 语言实现,在 Unix 工作站和 PC 机上运行。CADiZ 是 Z 规约的浏览和语法、类型检查工具,没有提供交互式的文档编辑工具,规约通过 LATEX 格式的 ASCII 文件进入系统,并提交给分析程序进行语法分析。可进行类型和作用范围检查。浏览工具以易读的打印格式显示规约,用户可在多个 Z 文档中进行搜索,可查询部分静态语义性质,如表达式的类型、引用模式的定义等。CADiZ 可以在文档中说明让 Z 模式横向排列显示,用户可交互查询规约中某一成分的性质,可在语法检查出错报告和 Z 规约之间进行双向查询。但 CADiZ 中未提供规约的编辑工具,只能对已按规定文本格式输入的 Z 规约文件进行图形方式浏览和语法、类型检查,从功能上讲不够完整,要与其他工具配套使用。证明工具可显示由用户产生的证明树,并已实现 W 逻辑规则,可辅助用户应用这些规则进行一些简单的证明。CADiZ 提供一个额外的工具 Zeta,实现将 Z 的规约精细化到 SPARK Ada 的代码。工具中支持为每一步推理步骤产生合适的验证条件,并将条件输入到相应的定理证明工具中,生成可编译的 SPARK 代码。大型的规约在 CADiZ 中已经得到过测试,但对内存的要求很高。CADiZ 没有提供对多用户使用的支持。由于系统具有很好的模块化机制,所以蕴藏了很好的可扩展性。

ForMooZ 是 UFPE/ITEP 在 Sun 4 工作站上用 Sather 和 C 开发的 MooZ 超文本环境支撑工具,包括采用图形界面的编辑、浏览工具和语法检查工具。编辑和浏览工具采用图形接口界面,对 MooZ 规约进行结构化的细粒度输入方式,每个定义都有单独的编辑窗口和浏览窗口,编辑窗口采用 LATEX 格式或 WYSIWYG 编辑器结合 LATEX 格式输入,输入正确的定义可在浏览窗口以图形方式浏览。这些窗口作为超文本结构的结点,各定义之间的联接由 MooZ 编译器自动创建,并可以按照不同种类对各定义进行分类排序,因此交叉浏览非常方便。编译器用 Unix 环境下的 *lex* 和 *yacc* 实现,可完成语法、类型和作用范围检查。ForMooZ 的另一特点是可供多人联网使用。由于每个模式都有单独的窗口,对于稍大的规约而言,ForMooZ 的窗口、对话框太多,界面过于复杂,不易了解规约的全貌。

Z/EVES 是可运行在 SunOS, OS/2, Linux, Windows 3.1/95/98/NT, Sun, Solaris 等多种操作系统上的 Z 规约的分析工具。它在支持对 Z 规约的分析上很有特点,提供了语法和类型检查、模式扩展、前置条件计算、域分析、符号执行和一般定理证明机制。对规约的精细化尚不支持。目前 Z/EVES 和 Z Browser 间的动态数据交换工具已经完成,可以使用 Z Browser 来编

辑浏览 Z 规约, Z/EVES 也支持大型规约的类型检查、模式扩展和域分析。

COOZ-Tools 是南京大学国家软件新技术重点实验室在 SGI 工作站上用 X-Windows 和 Motif 开发的支持 COOZ 的集成化工具环境 COOZ-Tools 1.0 版。该系统具有较好的可移植性,可以方便地移植到其它工作站或微机 UNIX 环境中,COOZ-Tools 是一个图形化的、内置 COOZ 语法分析工具的集成环境,支持建立和管理 COOZ 书写的规约文本。它主要由四部分组成:规约编辑、浏览工具、语法、类型检查工具,联机帮助工具,项目管理工具。通过转义符定义了所有符号的对应文本方式输入,通过符号浏览工具以图形方式显示规约,以符合用户的阅读习惯。用户可同时打开多个文本编辑窗口,每个编辑窗口除正常的编辑功能外,还提供了多种浏览工具:抽象显示、符号显示、规约结构和子类型关系树浏览工具。在编辑文档期间可随时检查输入的文本是否符合 COOZ 的语法,并产生有关错误的详细说明信息,提示用户错误的具体位置、性

质,以方便用户修改。用户可以很方便地查询到有关符号、语法等信息。联机帮助工具采用开放式接口,扩充极为方便,只需按简单的格式增加新的帮助文件即可。COOZ-Tools 可同时支持多个项目的开发,项目管理工具可辅助用户对各应用项目进行管理,其功能包括:项目的建立、维护;环境设计;文档管理;规约的重用等。COOZ-Tools 也有许多地方有待完善和提高,如基于模板的结构化编辑和所见即所得的图形化编辑功能、对规约中各成分的性质(如表达式的类型)的查询功能、用户界面的改进、网络功能等。COOZ-Tools 也没有提供定理证明和规约精化的功能。对大型规约的支持也较差。

Z 及其面向对象扩充的集成支撑环境的开发是一个研究趋势,现有的工具也在不断完善和扩充。其他一些 Z 辅助工具有 PiZA、TARDIS/Z、zedB、Zork、Z Browser 等,表 1 给出各种工具的综合比较。

结论 从本文对各种 Z 的支撑工具的比较可以看出,现有的 Z 支撑工具主要集中于规约的编辑和显

表 1 几种 Z 的支撑工具的评价与比较

比较内容	Formalser	ProofPower	Zola	CADiZ	ForMooZ	Z/EVES	COOZ-Tools
Z 的版本	标准 Z	标准 Z	Z 的扩展版本	标准 Z	MooZ	标准 Z	COOZ
编辑工具	语法制导	SunView 文本编辑	语法制导, 文本输入	LATEX 格式 ASCII 文件	LATEX 格式 WYSIWYG	借助 Z Browser	文本编辑
浏览工具	符号浏览, LATEX 格式	转化为 LATEX 格式	标准格式, 抽象, PS 格式	打印格式显示	图形显示	借助 Z Browser	图形显示
语法检查	支持	支持	支持	支持	支持	支持	支持
类型检查	支持	支持	支持	支持	支持	支持	支持
证明工具	不支持	支持	支持	简单的证明	不支持	支持	不支持
精化工具	不支持	不支持	部分支持	支持	不支持	不支持	不支持
对大型规约的支持	支持	限于中等规模	支持, 有局限	支持, 有局限	不支持	支持	不支持
多用户使用情况	不支持	不支持	支持	不支持	支持	不支持	不支持
工具的可扩展性	好	一般	不好	很好	一般	一般	好
使用平台	Windows	Sunview	Sunview 或 X Motif	Unix 或 PC	Sun4	SunOS OS/2 Linux Windows 等	X Windows 易于移植到其他平台

示、以及语法和类型检查方面,也有些工具支持定理证明和规约的推理精化,但工具都在定理证明上存在一定的缺陷或者有很大的局限性,规约的精化在工具上的支持仍然很少。在我们对 COOZ-Tools 进一步开发和研究中,将着重于定理证明、规约精化推理以及实现对大型规约的支持。

参 考 文 献

1 David Jordan Professor John McDermud, Dr Ian Toyn.

CADiZ-Computer Aided Design in Z

2 Flynn M, Hovard T, Formaliser D B. An Interactive Support Tool for Z

3 ORA Canada Z/EVES Version 1.5: An Overview. September 1998

4 Steggles P, Hulance J. Z TOOLS SURVEY. June 1994

5 袁晓东. Z 的面向对象扩充和程序转换系统的研究:[博士学位论文]. 1997. 12