

量子程序设计语言 NDQJava^{*}

徐家福⁺, 宋方敏, 钱士钧, 戴静安, 张云洁

(南京大学 计算机软件新技术国家重点实验室, 江苏 南京 210093)

Quantum Programming Language NDQJava

XU Jia-Fu⁺, SONG Fang-Min, QIAN Shi-Jun, DAI Jing-An, ZHANG Yun-Jie

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

+ Corresponding author: Phn: +86-25-83593468, Fax: +86-25-83593468, E-mail: xjf@nju.edu.cn, <http://www.nju.edu.cn/>

Xu JF, Song FM, Qian SJ, Dai JA, Zhang YJ. Quantum programming language NDQJava. *Journal of Software*, 2008,19(1):1-8. <http://www.jos.org.cn/1000-9825/19/1.htm>

Abstract: Since the first quantum programming language came about in 1996, many scientists and researchers are interested and involved in this field. After talking about several representative quantum programming languages, this paper gives an overview of the quantum programming language NDQJava designed by the authors, including design criteria, language paradigm, hardware platform, basic components and examples. Besides, the related works have also been mentioned.

Key words: quantum programming language; imperative quantum programming language; declarative quantum programming language; quantum data type; quantum variable; quantum expression; quantum statement

摘要: 量子程序设计语言自 1996 年出现以来, 颇受业界重视。在简述几种有代表性的量子程序设计语言之后, 着重阐述自行设计之量子程序设计语言 NDQJava 之概貌, 其中包括设计准则、语言风范、硬件平台、基本成分以及示例等。此外, 还提及相关工作。

关键词: 量子程序设计语言; 命令式量子程序设计语言; 申述式量子程序设计语言; 量子数据类型; 量子变量; 量子表达式; 量子语句

中图法分类号: TP301 文献标识码: A

1 沿革与动因

1.1 沿革

1982 年, Feynman^[1]提出了利用量子力学构建计算机的设想, 1985 年, Deutsch^[2]指出, 为量子计算机设计程序设计语言为一项有意义的工作。随着 1994 年 Shor^[3]关于大数质因子分解算法与 1996 年 Grover^[4]关于数据库搜索算法等划时代意义工作的出现, 1996 年出现了两种量子程序设计语言, 即 Knill 的量子伪码(quantum

* Supported by Natural Science Foundation of Jiangsu Province of China under Grant No.BK2007138 (江苏省自然科学基金)

Received 2007-05-18; Accepted 2007-06-30

pseudocode)^[5]与 Baker 的 Qgol^[6],以后,1998 年,Ömer 设计了 QCL^[7],2001 年,Zuliani 设计了 qGCL^[8],2004 年,Selinger 设计了 QPL^[9],2005 年,Altenkirch 等人设计了 QML^[10].上述工作途径不同,风范各异,堪称量子程序设计语言领域具有代表性的工作.

1.2 动因

2004 年起,笔者开始致力于量子计算的学习与研究,首先学习量子力学基本原理,特别是与量子计算密切相关之量子态叠加、量子测量以及量子纠缠等问题,继而学习了量子程序设计语言之基本文献.在此基础上,自行设计并在现有经典计算机上模拟实现了一种实验性的量子程序设计语言 NDQJava.从事此项研究之动因可归结为以下 3 点:第一、有感于量子计算机之诱人潜力,在其上求解某些问题,其复杂度远低于经典计算机;第二、虽然迄今实用的量子计算机尚未问世,但笔者相信,这只是时间问题,一旦量子计算机付诸实用,整个信息领域之面貌必将大为改观,从而对科学进步、社会发展以及人类文明等方面之推动将难以想象;第三、笔者研究经典程序设计语言多年,积累了一些经验与教训,对量子程序设计语言之研究将会有所借鉴.

2 几种有代表性的量子程序设计语言

2.1 量子伪码

Knill 在“量子伪码约定(conventions for quantum pseudocode)”一文中提出了书写量子伪码的几项约定,用作低级语言书写量子算法程序,并与量子计算中的量子随机存取机(quantum random access machine,简称 QRAM)相容.文中所提建议包括:提出量子寄存器概念以及区分量子寄存器与经典寄存器之标记方法,指明量子寄存器之纠缠范围的注释,以及量子寄存器之初始化、使用、测量等方法.此外,还提供了诸如测量除外的量子操作之取逆、条件量子操作,以及将经典算法转化成可逆算法等高级设施.作者指出,文中所提之约定尚属初步,有待完善.严格说来,作者提出之量子伪码约定尚难以称为一种量子程序设计语言.

2.2 Qgol

Baker 在““Qgol”:一种模拟量子计算的系统:理论、实现与洞察”(“Qgol”:A system for simulating quantum computations: Theory, implementation and insights)一文中指出,Qgol 是一种颇有意义的纯量子态处理语言,它是一种真正形象化(可视化)程序设计语言.系统之基本符号是作为量子对象之有向线路以及作为量子操作的门.模拟系统包括正文与图形相结合的编辑程序,该程序用函数式语言 Concurrent Clean 书写,运行于 Mac OS, Linux, OS/2 等操作系统之上.此外,文中还讨论了涉及通用量子计算机的一些基本问题.例如,量子模拟之非面向对象性等.

2.3 QCL

Ömer 设计的(1998 年~2003 年)QCL 语言是一种严格意义下的命令式量子程序设计语言.程序由一系列语句和定义组成.定义了标量、张量、量子 3 类类型,标量类型包括整型、实型、复型、布尔型和串型;张量类型包括向量、矩阵以及 n 阶张量 3 种;量子类型有通用寄存器、量子常量、目标寄存器、擦除寄存器几种.语句分简单语句(赋值、子程序调用、输入与输出、测量与初始化)、控制语句(循环、如果、异常终止)以及交互命令(模拟、列表、开壳、退出).定义有常量定义、变量定义、过程定义、操作符定义、量子函数定义、外部操作符定义、外部量子函数定义等.表达式可由字面信息、变量、常量等通过操作符与函数调用组合而成.该语言的各个成分均有严格的语法定义,并有明确的操作语义定义.惜其成分似嫌过多,背离了语言设计的简明性准则.作者在阐述该语言之前还明确指出,任何可用之量子程序设计语言所应具备的特性,如构造性、与硬件无关性、提供任意级之抽象以及在语义上集成多种非经典特征.此外,在设计量子程序设计语言时,必须反映量子计算之特点,如酉操作之可逆性、量子位之非定域性、状态之不可观察性、测量之破坏性以及擦除操作之阙如等等.这些内容对于任何一种量子程序设计语言之设计均至关重要.

2.4 QPL

Selinger 于 2004 年设计了一种函数式量子程序设计语言 QPL,其中引进了“量子流程图(quantum flow chart)”概念,致使程序表示更加结构化,有利于明确区分量子程序设计中数据流和控制流;QPL 的某些语法机制保证了所写程序在静态类型检查时即可发现错误(如违背量子不可克隆原理),从而避免了该类错误在运行时发生.同时,QPL 在语法和(初步的)操作语义上支持量子程序设计语言的某些高级成分(如循环、结构数据类型和递归过程调用等).QPL 目前仅见语言文本,尚未见其处理系统的实现.

2.5 QML

QML 是由 Altenkirch 和 Grattage 于 2005 年共同开发出的一种函数式量子程序设计语言,其中引进了量子数据结构与量子控制结构,并集成了可逆与不可逆量子计算.该语言基于严格线性逻辑,因之可能导致消相干之弱化必须显式给出.作者给出了语言的语法定义、操作语义以及指称语义.

QML 程序为一列函数定义.作者给出了 QML 函数定义之严格定义,其基础是类型和项:类型的定义是在定义了原始类型之后,借助张量积 \otimes 作为类型构造子来定义的;项的定义是名(受限或未受限)、项对、情形构造、函数等,文中使用了 Haskell 风格之语法以及其他一些特定记号,给读者带来不便,但其内容对量子程序设计语言之设计者来说却颇富启发性.

3 设计准则

3.1 实用性

设计语言的目的在于用.实用性作为语言设计的首要准则不言而喻.简言之,实用性指的是实际可行,用户满意.实际可行,其义自明,即指所设计之语言可在一定的设备(经典设备、经典与量子的混成设备、纯量子设备)上实现,并能书写多种量子算法,所有程序均能在相应设备上运行.判定一种语言之优劣有多种观点(如语言设计者观点、语言实现者观点、用户观点),但用户观点极具权威性,很难设想,一种用户不满意的语言会具有很大的生命力.此诸事例,在程序设计语言发展中屡见不鲜,于此不拟赘述.

3.2 简明性

简言之,简明性指的是内容简要、形式明了.申言之,简明性可包括以下 4 点:第一、区分内外性态.内部性态指的是一种语言构造所定义的成分只允许在该构造内部使用的一种性态;外部性态反之,即一种语言构造所定义的成分可以有条件地允许在该构造外部使用;第二,减少接口信息.接口问题是程序设计中既重要又难于处理的问题,在满足程序功能的前提下,宜尽可能减少接口信息.这一点对语言设计来说亦应予以考虑;第三,记号通用易读.语言定义中宜避免使用怪癖记号,在可能的范围内,应尽量使用常用记号,俾能便于阅读、便于理解、便于使用;第四,文档扼要流畅.语言文档要抓住要点、抓住本质,一种语言具有上千页之定义规约,不能不使人望而生畏,此种文档只有备查作用.因此,文笔宜流畅,避免晦涩行文.

3.3 严谨性

为了避免歧义,语言之语法应有形式定义,语言之语义亦然,但后者可以分两步走:第 1 步,先写出较为完整而准确的非形式定义,待语言初步实现后,再进行第 2 步,给出一种或多种形式语义;至于语用,由于目前所见工作不多,但严格说来,语用对用户说来颇为重要,盖因语言之语义乃语言成分固有之含义.这种含义与使用情景无关.语言之语用乃语言成分在使用情景下之含义,亦即和使用情景有关之含义,语用问题为语言研究者、设计者应予注意之重要问题.

3.4 快速性

古人云:“兵贵神速”,此中之速乃以决心取胜为前提,以正确之战略战术为前提.一句话,这里的快速性之确切含义是在确保质量之前提下,力争速度.质言之,根据本研究组具备之主客观条件,语言之设计宜在 2006 年 2

月前基本结束,盖因自 2006 年春季开始,即将开始讨论实现方案.当然,在实现方案之讨论以及往后之实现过程中,仍然会对语言中某些成分提出异议,届时再根据具体情况具体分析研究、改进之.

4 语言概况

4.1 语言风范

语言设计中遇到的首要问题是语言风范问题,即吾人设计之量子程序设计语言究竟是基于命令式风范抑或申述式风范,如所熟知,前者的物质基础是命令驱动式计算机,迄今使用之绝大多数计算机均属此类,因而与现有计算机相吻合.对用户说来,易学、易用、便于模拟实现.但其缺点是与量子算法之本质相悖(量子算法与经典算法不同,不一定能够明确指出其各个算法步之次序,因而可以说,量子算法本质上是申述式的),且不少命令式语言均具有函数副作用,程序不够紧凑优美;后者以数据(或其他非命令)驱动式计算机为其物质基础,理论基础为 λ 演算,与其物质基础相容,且与量子算法本质一致.另外,计算对象单次计值,无副作用,程序紧凑优美;其缺点是,对用户之数学素养要求较高,而且在经典计算机上模拟实现时功效较差.命令式语言与申述式语言各有利弊,本拟同时设计两种量子程序设计语言,待初步实现后,再根据实践进一步比较其得失,但限于主客观条件,无法同时并进,因而决定先进行命令式风范语言之实验.

4.2 硬件平台

虽然严格意义下之量子计算机目前尚未问世,但是不少人认为,开始出现的量子计算机可能是一种混成式的设备,即它由两部分组成:一部分是经典设备(经典计算机),另一部分是量子设备.前者负责控制与经典计算,后者负责量子计算,待处理的数据(指量子数据)先进入经典计算机,经过初态制备(即初化),进入量子设备.此后,在量子设备上对各种酉操作、张量积操作,然后进行测量操作,使终结量子态坍缩到所需之基态,将相应本征值作为计算结果送往经典设备.概言之,量子设备只进行量子计算,其他经典计算以及整个计算流程的控制均由经典设备完成.这样的量子计算机称为混成式量子计算机;即经典设备与量子设备协同工作的量子计算机,其中经典设备主理控制及经典计算,量子设备主理量子计算.这样,对算法设计者及程序设计者便提出以下要求,即在程序中宜尽可能地将经典计算与量子计算隔离,实在难以隔离时,则应另行设法弥补之.

4.3 语言结构

鉴于上述混成式之硬件平台,语言亦将采用相应的结构,语言 NDQJava 包括经典成分与量子成分,经典成分负责控制与经典计算;量子成分主要负责量子计算.考虑到 Java 语言颇为流行,故而语言之经典成分采用 Java 语言,量子成分自行设计,其原则是,既要考虑到只宜设置为数不多的为量子计算所必须的成分,同时又要考虑到易学易用.这里何谓为量子计算所必须的成分?理论上说,除测量外,语言之经典成分亦能表述量子计算.当然,这样做会大大降低功效,因而这里的“必须”乃是在不影响计算功效前提下的“必须”.在这种意义下,真正必须设置之量子成分相当有限,但是,兼顾到易学易用的原则,我们还是在“必须”与“易用”之间进行了适当的权衡.

4.4 量子成分

下文量子成分之表述一律为非形式表述,其形式表述详见另文.

4.4.1 量子数据类型 qtype

为了表述量子数据及其上的操作,定义了量子数据类型 qtype,其值集为量子态集,操作集为由酉操作、张量积操作以及测量操作所构成的集合.

其背景是,作为量子计算之量子数据是各种量子态;量子操作则是施于量子态之酉操作、张量积操作以及测量操作.

4.4.2 量子变量

量子变量是以 qtype 类型为类型之变量,表述为量子变量名(不受限的情形)或类名 \cdot 量子变量名或 this \cdot 量子变量名(受限情形)(其中,类名表示该变量在其中定义的类的类名,this 表示该变量在其中定义的类).量子变量

说明形如

qtype 量子变量名,

其中量子变量名为标识符.

4.4.3 量子表达式

为简便计,定义了 4 类量子表达式,即预初始化表达式、酉表达式、张量积表达式以及测量表达式.

预初始化表达式用以对量子变量赋初态进行准备,其形如

表达式 :: 表达式.

这里的两个表达式均为取非负整值的经典表达式,第一个表达式的值表示所说明之量子变量使用的量子位数,第二个表达式表示在量子位数确定后量子变量将被初始化到的基态之序号.

酉表达式是量子变量或酉表达式施以酉操作的表述,其形如

量子变量名 酉操作符(实参表)

或

酉表达式 酉操作符(实参表).

这里的酉操作符有 $u_i, u_x, u_h, u_s, u_z, u_{rotx}, u_{roty}, u_{rotz}, u_{cnot}, u_{crotx}, u_{croty}, u_{crotz}, u_{swap}, u_{cpha}$, 共 14 种, u_i 表示对量子变量的某一位作 I 变换, u_x 表示对量子变量的某一位作 X 变换, u_h 表示对量子变量的某一位作 H 变换, u_s 表示对量子变量的某一位作 S 变换, u_z 表示对量子变量的某一位作 Z 变换, u_{rotx} 表示对量子变量的某一位绕 X 轴旋转某一弧度,参数表中第 1 个实参值表示进行旋转之位号,第 2 个实参值表示所旋转的弧度值, u_{roty}, u_{rotz} 之含义类似, u_{cnot} 表示对量子变量的某两位作 $CNOT$ 变换,第 1 个实参值为控制位号,第 2 个实参值为变换位号, u_{crotx} 表示对量子变量的某两位作沿 X 轴的控制旋转变换,第 1 个实参值为一字串,用以标定哪些位是控制位以及这些控制位的控制状态(T 或 F),第 2 个实参值为变换位号,第 3 个实参值为旋转之弧度值, u_{croty}, u_{crotz} 之含义类似, u_{swap} 表示对量子变量的某一位作 $SWAP$ (对换)变换,第 1 个实参值与第 2 个实参值分别表示作 $SWAP$ 变换的两个量子位的位号, u_{cpha} 表示根据量子变量的某些位作控制全局相位变换,第 1 个实参值为一字串,用以标定哪些位是控制位以及这些控制位的控制状态(T 或 F),第 2 个实参值为相位移动的弧度值.

张量积表达式是量子变量(或张量积表达式)与量子变量(或张量积表达式)之间施以张量积操作的表述,其形如

量子变量或张量积表达式 张量积操作符 量子变量或张量积表达式,

即

量子变量或张量积表达式 q_tensor 量子变量或张量积表达式.

测量表达式是对量子变量进行测量(全测量或部分测量)的表述,其形如

量子变量 $q_measure()$

或

量子变量 $q_measure$ (实参).

前者表示全测量,后者表示部分测量,其中实参值为应作测量的位号.

4.4.4 量子语句

有 4 类量子语句,即量子预初始化语句、量子赋值语句、量子测量语句以及量子分程序.

量子预初始化语句用以对说明中的量子变量进行初始化准备,其形如

量子变量:= $init$ 预初始化表达式.

量子赋值语句用以对量子变量赋值,所赋之值有 3 种情形:一为量子变量之值,二为酉表达式之值,三为张量积表达式之值.

使用量子赋值语句须满足以下二条件之一:一为所赋变量或表达式之值为基态;二为若变量或表达式之值非基态,则该值之经历足迹已知.

量子测量语句用以执行其中测量表达式,并将其值赋与其左方变量(该变量为经典变量).

4.4.5 量子分程序

设置量子分程序的目的在于使程序中的量子部分相对集中,这样既可以提高程序的易读性,又便于处理程序对经典部分与量子部分分别处理.

量子分程序是一组括以 `begin` 及 `end` 的量子变量说明或(与)量子语句.

4.4.6 程 序

本语言程序的定义与 Java 类似,其不同之处有二:一为在类说明中可有量子变量说明;二为有量子分程序.执行时需指定其一类中的 `main` 方法为程序之入口.

5 示 例

例 1:

```
public class Toss
{
    public static void main (string [] args)
    {
        Toss t=new Toss();
        int i=t.toss();
        System.out.println ("The result of Toss is"+i);
    }
    public int toss()
    {
        int i=0;
        begin
            qtype coin;
            coin:=init 1::0;
            coin:=coin u_h(0);
            i:=coin q_measure();
        end
        return i;
    }
}
```

例 2:

```
public class Deutsch
{
    public static void main (String [] args)
    {
        Deutsch d=new Deutsch();
        int i=d.deutsch_1();
        System.out.println ("The result 1 of Deutsch is:"+i);
        i=d.deutsch_2();
        System.out.println ("The result 2 of Deutsch is:"+i);
    }
    public int deutsch_1()
    {
        int i=0;
```

```

begin
    qtype qs;
    qs:=init 2::0;
    qs:=qs u_i(1)u_x(0);
    qs:=qs u_h(1)u_h(0); //输入函数 f 为平衡函数
    qs:=qs u_cnot(1,0);
    qs:=qs u_h(1)u_i(0);
    i:=qs q_measure(1);
end
return i;
}
public int deutsch_2()
{
    int i=0;
    begin
        qtype qs;
        qs:=init 2::0;
        qs:=qs u_i(1)u_x(0);
        qs:=qs u_h(1)u_h(0); //输入函数 f 为常函数
        qs:=qs u_i(0)u_i(1);
        qs:=qs u_h(1)u_i(0);
        i:=qs q_measure(1);
    end
    return i;
}
}

```

6 相关工作

6.1 语言风范

本语言为命令式语言.如前所述,命令式语言和申述式语言各有千秋.前述之 QML 即为申述式语言(函数式语言),我等即将设计之量子程序设计语言 NDQFP 亦为函数式语言.待实现后,再行比较.

6.2 语言成分

QCL 等语言中引进了多种量子类型(如通用寄存器,量子常量等)和一些较为高级之成分(如循环等).此诸成分有利于程序之编写与阅读,有待于在今后的实验中考虑取舍.至于模块,对于书写大型程序确为必要,但在量子程序设计语言中如何引进,尚待仔细斟酌.

6.3 语言语义

在本语言中仅给出其非形式语义,在 QCL 中定义了操作语义,QML 给出了操作语义与指称语义.严格说来,给出语言之形式语义确有必要.我等在第 2 个实验中即将考虑.

6.4 硬件平台

本语言以一种混成式量子计算机为假想之硬件平台,实际上是在经典计算机上模拟实现的(详见文献[10]),结果只能说明其可行性,难臻实用.今后拟与国内外从事量子硬件之合适单位合作,共同探讨在真正量子设备上实验之问题.

NDQJava 语言是本小组进行的首次尝试,它已于 2006 年 6 月底在本实验室的经典计算机上模拟实现,正确运行了几种量子算法.结果表明,达到了预期效果.然而,一则量子程序设计语言为一新兴领域,历史仅有 10 年,二则这是我们的初步实验,肯定存在不足甚至谬误之处,敬请读者批评指正.

致谢 参加本项工作的还有刘吉、吴楠、钱辰、焦阳、徐明君、董青、朱晓端、颜仙乐等人.中国科学技术大学张永德教授、南京有线电厂徐焕亮高级工程师等人均曾提出指导性意见,北京大学、国防科学技术大学、中创软件公司、苏州大学、河海大学、南京大学等单位均曾对本项目给予资助,在此一并深致谢意.

References:

- [1] Feynman R. Simulating physics with computers. *Int'l Journal of Theoretical Physics*, 1982,21(6):467-488.
- [2] Deutsch D. Quantum theory, the Church-Turing principle and the universal quantum computer. *Proc. of the Royal Society A*, 1985,400(1818):97-11.
- [3] Shor PW. Algorithms for quantum computation: Discrete log and factoring. In: *Proc. of the 35th Annual Symp. on the Foundations of Computer Science*. Santa: IEEE Computer Society Press, 1994. 124-134.
- [4] Grover LK. A fast quantum mechanical algorithm for database search. In: *Proc. of the 28th Annual ACM Symp. on the Theory of Computing*. ACM Press, 1996. 212-219.
- [5] Knill EH. Conventions for quantum pseudocode. LANL Report, LAUR-96-2724, 1996.
- [6] Baker GD. "Qgol": A system for simulating quantum computations: Theory, implementation and insights [Honours Degree Thesis]. Macquarie University, 1996.
- [7] Ömer B. A procedural formalism for quantum computing [MS. Thesis]. Vienna: Technical University of Vienna, 1998.
- [8] Zuliani P. Quantum programming [Ph.D. Thesis]. Oxford: University of Oxford, 2001.
- [9] Selinger P. Towards a quantum programming language. *Mathematical Structures in Computer Science*, 2004,14(4):527-586.
- [10] Song FM, Qian SJ, Dai JA, Zhang YJ, Xu JF. Processing system of quantum programming language NDQJava. *Journal of Software*, 2008,19(1):9-16 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/19/9.htm>

附中文参考文献:

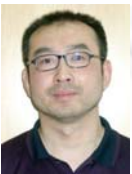
- [10] 宋方敏,钱士钧,戴静安,张云洁,徐家福.量子程序设计语言 NDQJava 处理系统.软件学报,2008,19(1):9-16. <http://www.jos.org.cn/1000-9825/19/9.htm>



徐家福(1924—),男,江苏南京人,教授,博士生导师,CCF 高级会员,主要研究领域为高级语言,新型程序设计,软件自动化.



戴静安(1980—),男,硕士,主要研究领域为计算机软件.



宋方敏(1961—),男,教授,博士生导师,CCF 高级会员,主要研究领域为计算机软件与理论.



张云洁(1984—),女,硕士生,主要研究领域为计算机软件.



钱士钧(1941—),男,教授,主要研究领域为计算机软件.