

美国2001'计算教程综述^{*}

Overview of Computing Curricula 2001

贲可荣 肖斌

(海军工程大学计算机系 武汉430033)

Abstract This paper reviews Computing Curricula 1991, introduces the background and the principles of Computing Curricula 2001, which is developed by the Computer Society of the Institute for Electrical and Electronic Engineers (IEEE-CS) and the Association for Computing Machinery (ACM), lists the main contents of CC2001.

Keywords Computer education, Computing curricula

1. Computing'91教程回顾

1991年,美国计算学科主要的专业学会 ACM 和 IEEE-CS 联合颁布了 Computing'91教程,该教程力图覆盖计算学科的计算机科学、计算机工程、计算机科学与工程等各类本科生专业。Computing'91教程的提出对国内外计算机教育界产生了较大影响,该教程的系统性、完整性、合理性和灵活性均受到普遍的称赞。中国计算机学会教育委员会和全国高等学校计算机教育研究会也联合制定了类似的教程:“计算机学科教学计划1993”。Computing'91教程第一次对计算学科给出了透彻的定义:计算学科系统地研究信息描述和变换的算法,研究算法的理论、分析、设计、效率、实现和应用。一切计算的基本问题是“什么能被有效地自动化?”计算学科具有科学与技术的双重性。显然“计算”是广义的,并用这个术语来统一地指称现行的各类计算机专业。该定义是科学的、动态的,反映了学科的共性及迅速发展特点。按照该定义,教程引申出了九个核心的主干科目领域,并规定其作为基础的教学内容。Computing'91教程还用“理论、抽象、设计”三个过程来强调各部分内容的教学方法及形式的多样性。Computing'91教程也提出了重复出现的十二个概念,用以揭示学科的内在线索。Computing'91教程的灵活性主要反映在它并不直接规定课程,而是定义核心的知识单元及教学方法^[5]。

Computing'91主干科目知识单元:

(1)算法与数据结构(AL)。内容包括:基本数据结构;抽象数据类型;递归算法;复杂性分析;复杂性分类;排序和搜索;可计算性与不可判定性;问题要求策略;并行与分布式算法。

(2)计算机体系结构(AR)。内容包括:数字逻辑;数字系统;数据表示与转换;处理器结构与汇编;存储系统结构;输入输出系统结构;其它体系结构。

(3)人工智能与机器人学(AI)。内容包括:人工智能概论;问题状态及搜索;专家系统;机器人学。

(4)数据库与信息检索(DB)。内容包括:数据库概论;模型及应用;关系数据库。

(5)人机通信(HU)。内容包括:用户界面和设备;计算机图形学。

(6)数值及符号计算(NU)。内容包括:数值及符号计算;数的表示、误差及传播;逼近与迭代方法;科学计算中的常用算法。

(7)操作系统(OS)。内容包括:操作系统历史及演变;任务和进程;进程调度及同步;处理机管理;存储管理;设备管理;文件系统;安全与保护;通讯与网络;分布与实时处理。

(8)程序设计语言(PL)。内容包括:程序语言历史;虚拟机;数据类型;数据结构;数据控制、共享和类型;运行存储管理;有限自动机和正规表达式;文法和自动机;语言翻译系统;语言语义;非过程语言;分布和并行程序设计。

(9)软件方法学与软件工程(SE)。内容包括:问题求解的基本概念;软件和软件工程;软件需求及规格说明;软件设计和实现;软件验证和确认;软件开发计划和管理;软件开发工具与环境。

2. CC2001产生的背景

Computing'91对世界计算机教育产生了巨大的影响,随着计算机技术的飞速发展,人们不能满足于 Computing'91,认识到:(1)知识单元在实际教学大纲

^{*}教育部高等学校骨干教师资助。贲可荣 博士,教授,主要研究方向:软件工程,软件可靠性,人工智能。

制订中不如课程设计操作性强。(2)针对各个知识单元,应规定哪些知识是必不可少的,(3)应制定一些适用于不同层次、不同特点的学校的示范性教学大纲,(4)教程应更加注重计算机科学和计算工程大纲的界定准则。

进入新世纪,计算是一个充满活力的领域。计算技术已成为时代的标志技术,计算机成为现代文化的组成部分,成为世界经济发展的主要推动力,这一领域正继续以惊人的速度发展着。新的技术不断引入,而旧的技术几年之内就很快过时。

学科的飞速发展对计算教育的内容和教学有着深远的影响。例如,当 Computing'91 制定时,网络技术并未当作一个主要知识单元,只有6学时的一般要求内容。对网络缺乏重视并不令人惊讶,毕竟当时网络还没有广泛的市场,万维网与其创造者的设想也没什么差别。今天,仅仅十年之后,网络技术和互联网成为我们许多经济的支柱。它们成为计算领域的基础,难以想象大学教学计划怎能不为这一知识单元腾出相当多的学时。同时,互联网的存在也改变了教育过程的性质,现代网络技术增强了每个人相互交流的能力,为全世界的人们提供了空前的获取信息能力。现在大多教学计划——不光是计算,其它专业也一样——网络技术已经成为一个基本的教学工具。

在过去十年,计算有着很大的变革。这些变革分为两类:技术的和文化的。任何一类变革都给计算教学带来了显著的影响。

2.1 技术的变化

许多影响计算的变革来自技术的进步。很多技术进步是延续多年发展过程的一部分,1965年由 Intel 公司创始人戈登·摩尔预言的摩尔定律:微处理器的集成度每十八个月将翻一倍,继续得到验证。结果我们看到,计算机的性能以指数增长,使几年前还让我们无能为力问题得以解决。其它学科的变化(例如,互联网出现后网络的飞速发展),更富有戏剧性,说明了变化同样发生在革新阶段。无论是渐渐变化还是革命性变化都影响着计算和教育过程要求的知识主体。

过去十年技术的进步增强了许多知识单元的重要性,例如:互联网及其应用;网络技术,特别是其于 TCP/IP 的技术;图形学与多媒体;嵌入式系统;关系数据库;互操作性;面向对象程序设计;复杂应用程序接口的应用;人-机交互;软件安全性;保密与密码学;应用领域。

由于这些知识单元的增长,有必要将它们作为大学生要求的内容包括在内。不幸的是,在大多数学位教育方案中很难在不去掉旧内容的情况下加入新的知识单元。想覆盖新的领域而又不减少传统知识单元的学时几乎不可能。有些传统课程(如汇编程序设计、形式化语义学和数值分析)已经变得越来越不重要了。

2.2 文化的变化

计算教育同样受到文化及社会变化的影响。例如下面这些变化对教育过程均产生影响:

- 由新技术带来的教学方法的变化。推动近来计算技术进步的技术变革直接影响到教育文化,例如计算机网络,使远程教育更灵活,促使这一领域有了长足的发展。这些网络也使广泛分布的院校间能够更方便地分享课程素材。技术也影响教学手段,演示软件、计算机投影和个人实验台形成了计算教学的独特方式。计算教程的设计必须将这些正在变化的技术考虑在内。

- 全球计算技术的戏剧性增长。计算在过去十年中有着惊人的扩充。例如,在1990年只有很少的家庭接入了互联网(即使在美国也一样)。一个美国商业研究组织透露:到1999年超过三分之一的美国人使用互联网,在其它国家同样有类似的增长模式。对计算技术使用的激增带来了许多影响教育的挑战,包括学生通过计算机相互熟悉普遍增多,使用过计算机和未使用过计算机者之间差距不断加大,计算技术的应用在增长。

- 计算技术正在增长的经济影响力,围绕着高科技产业的戏剧性活力,正如互联网过去五年的火爆说明,对教育及其可用资源有重要的影响。对计算机专业人员的巨大需求以及对广阔未来的展望吸引了更多的学生来到这一领域,甚至包括一些对此并不感兴趣的人。同时,工业的需求使大多数院校更难吸引和挽留其教员,为院校满足需求的能力产生了很大制约。

- 计算作为学术学科被更广泛地接受。最初,在许多院校中计算不得不努力争取或为主干学科。毕竟当时它不是一门支持大多数学术领域的有历史基础的学科,在某种程度上,这个问题在制定与《计算作为一门学科》的报告紧密联系的 CC1991 时也存在^[1]。计算技术已成为文化和经济的组成部分,作为一门学科存在的合理性已毋庸置疑。在许多学校,计算成为了最大和最有活力的学科之一,再不需要为了让院校加入计算教育而争辩,今天的问题是找到满足需求的方法。

- 学科的拓展。在我们的学科发展和成为主干学科的同时,它也拓展了范围,在早期,计算主要集中在以数学和电子工程为基础的计算机科学上。一段时间后,越来越多的领域组成了更大、包容更多内容的计算学科。

3. 制定 CC2001 的原则

基于我们对过去教程报告和学科大纲的变化的分析,CC2001工作组明确提出以下原则:

(1)计算成为一门极为广阔的学科,远远超出了传统计算机科学的界限。由于近些年由计算机科学演化而来的子学科,不能再简单地将计算课程归入计算机科学。学院和大学必须对这些新领域的出现有所敏

感,确保基础计算课程有广泛的听众。

(2)尽管它的增长幅度很大,计算仍是一个从很多完善的学科中建立起来的综合研究领域。在它的所有子学科中,计算建立在数学、科学、工程、心理学、管理学以及其他领域的基础之上,这些领域都要求理论与实践相结合。正如 Computing'91报告中指出的“对学科的掌握不仅包括对基本课程的理解,而且包括这些概念对现实问题的可应用性的理解”。特别应该注意在大学教程中实验工作的重要性,因为这样做使学生进一步掌握了概念及其在不同领域中解决实际生活问题的应用。

(3)计算学科快速的演变要求对相应教程要有发展的眼光。本学科按此变化步伐,每十年更新一次教程是不行的。学科的专业联系必须建立在不断进行的教程评审上,允许教程的个别部分循环更新。

(4)CC2001必须超越知识单元而针对个别课程设计提供重要的指导。虽然 Computing'91中的知识单元结构可以作为有用的框架,但大多数院校需要更多的详细指导。对于这些院校来说,CC2001定义一可选模型集合(大概2到4个),将知识单元结合成合理的、易于实施的课程,它在某种程度上是有效的。明确表达一组定义完善的模型集合会使院校共享教学对策和工具变得更容易。同时也可以为向这些课程提供课本和其它资料的出版社提供一个框架。

(5)CC2001必须确定一组相对小的所有计算专业学生必需的核心概念和技能。从历史来看,学科的发展都导致计算机科学内核的平行扩展。当重要的新单元出现时,都有很强的诱惑包含它们,正如大学生所需要的那样。在过去十年,学科扩展到很难简单地加入新单元而不去掉其它单元的程度。由于大学生学位的限制,很难加入软件工程、人机交互、网络和图形学的新内容,还以传统方式保持全部传统的经典内容,如:汇编语言程序设计,编译器构造和自动机理论。最好的内容是减少必需的计算机科学核心的大小,允许有更大的灵活性来加入新的单元适应教程的变化。CC2001工作组一致同意“核心由公认的对于大学生学位来说是基本的单元(包括计算机科学,计算机工程和其它类似名称的课程)组成”。这一定义必须包含对所有的大学课程都通用的基本要求。同时,核心不是靠自己制定一个完全的大学教程,而是必须有依据院校、研究领域和学生个体而变化的附加课程的支持。

(6)CC2001必须为核心课程之外的课程提供指导。在确定学科基础核心之外,CC2001必须提供为更高级领域中的高级技术选修课提供指导。

(7)CC2001必须是国际范围的。CC2001的预期指导范围不仅仅限于美国,而是希望对全世界计算专业

都有所帮助。其它国家课程的要求经常和美国有很大区别。

(8)CC2001的发展必须有工业界的积极参与,绝大多数计算机专业大学毕业生都在工业部门工作,常常受不到更高层次的教育,为了确保毕业生能适应这些岗位要求,相信在设计、开发和落实新课程中有实业界人士的参与是必要的。

(9)CC2001必须将职业实践作为一个完整的部分加入大学教程。因为计算是一个综合的学科,强调学科理论的同时强调实践是必要的。今天,许多与计算有关的实践知识以职业实践形式存在于工业中。为了在那些环境中成功地工作,学生必须接受那些作为教育一部分的实践。并且,这些实践超出了计算技能,包含了管理、伦理和价值、写作和口头表达以及团队协作能力等更广泛范围的知识。

(10)CC2001必须努力做到对它所期望的听众有用。为了保持可用性,CC2001必须支持由计算机科学专家委员会和工程技术专家委员会以及美国类似组织认可的大纲;CC2001必须具备充分的一般性以满足各种不同重点和不同目标的绝大多数计算大纲的需求;CC2001必须足够灵活以适应未来计算学科的发展。

4. CC2001教程知识单元

CC2001教程包含14个知识单元:

(1)离散结构(DS)(37核心学时) 内容包括:函数,关系和集合(6);基本逻辑(10);证明技术(12);计数基础(5);图和树(4)。

(2)程序设计基础(PF)(65核心学时) 内容包括:算法和问题求解(8);基础程序设计构造(10);基本数据结构(12);递归(6);抽象数据类型(9);面向对象程序设计(10);事件驱动和并发程序设计(4);使用现代应用程序接口(API)(6)。

(3)算法和复杂性(AL)(31核心学时) 内容包括:基本算法分析(4);算法策略(6);基础计算算法(12);分布式算法(3);基本可计算性理论(6);复杂性类P和NP;自动机理论;高级算法分析;加密算法;几何算法。

(4)程序设计语言(PL)(5核心学时) 内容包括:程序设计语言的历史和综述(2);虚拟机(1);语言翻译介绍(2);语言翻译系统;类型系统;执行控制模型;声明、模件性和存储管理;程序设计语言语义;函数程序设计范型;面向对象程序设计范型;基于语言的并发构造。

(5)体系结构(AR)(33核心学时) 内容包括:数字逻辑和数字系统(3);数据的机器层资料表示(3);汇编层机器组织(9);存储系统组织(5);I/O和通信(3);

CPU 实现(10)。

(6)操作系统(OS)(22核心学时) 内容包括:操作系统原理(2);并发(6);调度和发送(3);虚拟内存(3);设备管理(2);安全和防护(3);文件系统和命名(3);实时系统。

(7)人机接口(HI)(3核心学时) 内容包括:人机接口原理(3);用户建模;交互;窗口管理系统设计;帮助系统;评价技术;计算机支持的协同任务。

(8)图形学、可视化和多媒体(GR)(无核心学时) 内容包括:图形系统;图形技术;基本渲染;基本几何建模;可视化;虚拟现实;计算机动画;高级渲染;高级几何建模;多媒体数据技术;压缩和解压;多媒体应用和内容编辑;多媒体服务和文件系统;网络及分布式多媒体系统。

(9)智能系统(IS)(10核心学时) 内容包括:智能系统的基本论(2);搜索和优化方法(4);知识表示和推理(4);学习;智能代理;计算机视觉;自然语言处理;模式识别;高级机器学习;机器人学;基于知识的系统;神经网络;遗传算法。

(10)信息管理(IM)(10核心学时) 内容包括:数据库系统(2);数据建模和关系模型(6);数据库查询语

言;关系数据库设计;事务处理;分布式数据库;高级关系数据库设计;物理数据库设计。

(11)网络中心计算(NC)(15核心学时) 内容包括:网络中心计算介绍(9);互联网(Web)客户/服务器计算的实例(6);建造互联网应用;通信和网络;分布式对象系统;协作技术和群件;分布式操作系统;分布式系统。

(12)软件工程(SE)(30核心学时) 内容包括:软件处理和度量(9);软件需求和规范(6);软件设计和实现(6);验证和确认(6);软件工具和环境(3);软件项目方法学(3)。

(13)计算科学(CN)(无核心学时) 内容包括:数值分析;科学可视化;科学计算体系结构;并行结构程序设计;应用。

(14)社会的、伦理的和职业的论题(SP)(16核心学时) 内容包括:计算历史(1);计算的社会背景(2);分析的方法和工具(2);职业和伦理的职责(2);安全关键系统的风险和职责(2);知识产权(3);隐私和公民权力(2);Internet 的社会影响(2);计算机犯罪;计算中的经济论题;伦理学的哲学基础。

(下转第95页)

(上接第122页)

$$R_{Cq} = P_{Cq} \times C < r$$

$$P_{Cq} < r/C$$

由式(7)得:

$$S_q = 1 - P_{Cq} > 1 - r/C$$

上式表明了程序在第 q 次安全运行的概率要求。

定义5 设 C 为系统事故最大的后果严重性参数, r 为软件的可接受风险水平, 且软件的每次运行是不独立的, 则当软件第 q 次运行处在安全状态的概率 S_q 满足 $S_q > 1 - r/C$ 时, 称该软件为第 q 次运行安全性合格产品。

程序连续运行 m 次, 在前 $n-1$ 次运行正常, 第 n 次运行出现失效的概率为:

$$P_{n/n-1} = P_{Cn} \prod_{k=1}^{n-1} (1 - P_{Ck})$$

对应的风险为:

$$R_{(n/n-1)} = P_{n/n-1} \times C = P_{Cn} \prod_{k=1}^{n-1} (1 - P_{Ck}) \times C$$

如要求安全运行 $n-1$ 次后, 软件的风险仍处在可接受状态, 则要求满足条件

$$P_{Cn} \prod_{k=1}^{n-1} (1 - P_{Ck}) < r/C$$

定义6 设 C 为系统事故最大的后果严重性参数, r 为软件的可接受风险水平, 且软件的每次运行是

不独立的, 则当软件连续安全运行 $n-1$ 次, 在第 n 次仍保持安全运行的概率 $S_{n/n-1}$ 满足:

$$S_{n/n-1} = 1 - P_{n/n-1} = 1 - P_{Cn} \prod_{k=1}^{n-1} (1 - P_{Ck}) > 1 - r/C$$

时, 称该软件为 $(n-1)$ 次安全运行条件下的下一次运行安全性合格产品。

定义3~6分别是对应于一次性独立运行任务、连续 m 次独立运行任务、第 q 次非独立运行任务和在前 $n-1$ 次安全运行条件下运行第 n 次任务时的软件安全性评估准则。

参考文献

- 1 Bob M. Software in Safety-related System: Basic Concepts and Concerns. In: Phil Bennett, ed. Safety Aspects of Computer Control. Butterworth Heinemann Ltd., 1993. 1~18
- 2 鄢萌. 计算机软件的可靠性. 北京: 国防工业出版社, 1988
- 3 蔡开元. 软件可靠性工程基础. 北京: 清华大学出版社, 1995
- 4 Lyu M R. Handbook of Software Reliability Engineering. Los Alamitos, California. IEEE Computer Society Press, 1996
- 5 Nelson E. Estimating Software Reliability from Test Data. Microelectronics & Reliability, New York, Pergamon, 1978. 67~74

包分片时,就可能检测不出检测的内容。

“欺骗”检测的原理是使 NIDS 截获到的信息与接收方正常收到的信息不一致。

很多欺骗方法是和具体的应用层数据内容有关,通过发送假数据,使 NIDS 不能正确识别出检测方的真实数据,NIDS 的检测原理是分析各个应用的数据,看其是否符合入侵特征。例如,提供 TELNET 服务的主机,限制“root”用户远程访问,当 NIDS 检测到“root”用户试图远程访问时,就认定为“入侵”,针对这个检测功能,可以通过下面的方法进行“欺骗”检测。

假设 IP 包从检测点到接收方的生存期(Time To Live, TTL)为 20,从检测点到 NIDS 的 TTL 为 10。第一步,检测点先发送一个 IP 包 A 给接收方,TTL 等于 21,序列号为 100~103,数据内容是“USER”,NIDS、接收方都接收到该包。第二步,检测点再发送一个 IP 包 B 给接收方,TTL 等于 15,序列号等于 104~107,数据内容为“HACK”,该包可以到达 NIDS 所在的网络,被 NIDS 截获到,但不能到达接收方所在网络。第三步,检测点再发送一个 IP 包 C 给接收方,该包的 TTL 等于 21,序列号同 IP 包 B,为 104~107,数据内容为“root”,接收方和 NIDS 都能接收到这个包。最后,接收方收到的信息是“USER root”,NIDS 收到信息“USER hack”或“USER root”,如果 NIDS 接受 IP 包 B,忽略 IP 包 C,则收到信息“USER hack”;如果用 IP 包 C 覆盖 IP 包 B,则收到信息“USER root”。通过改变 IP 包 B 和 IP 包 C 的发送顺序,可以使 NIDS 只得到数据“USER hack”,成功欺骗 NIDS。

在“欺骗”检测中,除通过生存期来检测外,还可以通过改变 IP 包的校验值或利用网络的 MTU (maximum transmission unit)等方法来实现“欺骗”检测。

结论 目前,国内对 NIDS 的研究主要集中在实现 NIDS 的功能和提高性能上,对 NIDS 本身的安全性研究较少,NIDS 在安全系统中的重要程度还取决于其本身的安全性能,因此有必要对其进行检测,来验证它的安全性、可靠性。本文仅对检测 NIDS 本身的

安全性能提供了几种方法,还有待进一步研究。针对具体的 NIDS 产品来说,还需要从功能、价格等方面考虑,按照 2.4 节中列出的特点进行检测,如从检测类型设置、系统配置、报警、日志、结构、价格等因素来考虑。

参考文献

- Mukherjee B, Heberlein T I, Levitt K N. Network intrusion detection. IEEE Network, 1994, 8(3): 36~41
- Crosbie M, Sparrow G. Active defense of a computersystem using autonomous agents [Technical Report 95-008] CCAST Group, Department of Computer Sciences, Purdue University, West Lafayette, IN 47907-1358, Feb. 1995
- Stanford-Chen S. Common Intrusion Detection Framework. <http://seclab.cs.ucdavis.edu/cddi/>
- Stanford-Chen S, et al. GridS-A Graph-Based Intrusion Detection System for Large Networks. In: The 19th National Information Systems Security Conf., 1996
- Fox K L, et al. A Neural Network Approach towards Intrusion Detection. In: Proc. of the 13th National Computer Security Conf., Oct. 1990
- Pukeza N F, et al. A Methodology for Testing Intrusion Detection Systems. IEEE Transactions on Software Engineering, 1996, 22: 719~729
- StJohns M. Authentication Server. RFC 931, TPSC, Jan. 1985
- Stevens W R. TCP/IP Illustrated, Vol 1. Addison-Wesley, Reading, MA, 1994
- Postel J. Internet Protocol-DARPA Internet Program Protocol Specification. RFC 791, USC/Information Sciences Institute, Sep. 1981
- Postel J. Internet Protocol-DARPA Internet Program Protocol Specification. RFC 791, USC/Information Sciences Institute, Section 3.2, line 1099, Sep. 1981
- Atkinson R. Security Architecture for the Internet Protocol RFC 1825, Naval Research Laboratory, Aug. 1995
- Postel J. Transmission Control Protocol-DARPA Internet Program Protocol Specification. RFC 753, USC/Information Sciences Institute, Sep. 1981
- Jacobson V, Braden R, Borman D. TCP Extensions for High Performance. RFC 1323, IBL, ISI, Cray Research, May 1992
- Joncheray L. A Simple Attack Against TCP. In: 5th USENIX UNIX Security Symposium, June 1995. ICSA lab
- Paxson V. Bro, A System for Detecting Network Intruders in Real-Time. In: 7th Annual USENIX Security Symposium, Jan. 1998
- Paxson V. End-to-End Internet Packet Dynamics. In: ACM SIGCOMM'97, Cannes, France, Sep. 1997

(上接第 129 页)

致谢:感谢国防科技大学软件教研室主任王毅博士提供了有关资料。

参考文献

- Computing Sciences Accreditation Board. Criteria for accrediting programs in computer science in the United States. Technical report in preparation for June 2000. Version 1.0, January 2000. http://www.csab.org/criteria3k_v10.html
- Denning P J, Comer D E, et al. Computing as a discipline

Communications of the ACM, 1989, 32(1): 9~23

- Denning P J. Our seed corn is growing in the commons. Information Impacts Magazine, March 1999. <http://www.cisp.org/imp/march 99/denning/03-99denning.htm>
- Lidtke D K, Stokes G E, et al. ISCC'99: An information systems-centric curriculum' 99, July 1999. <http://www.iscc.unomaha.edu>
- Tucker A B, Barnes B H, et al. Computing Curricula' 91. Association for Computing Machinery and the Computer Society of the Institute of Electrical and Electronics Engineers, 1991