

模型检测;CTL * 模型检测。

重点及难点是模型检测算法;构建 tableau 结构。

14. 符号模型检测

主要包括:二叉判定图(BDD)及其约简;基于 BDD 的 CTL 符号模型检测;符号模型检测中的公平性;基于 BDD 的 LTL 符号模型检测;基于 SAT 的 LTL 符号模型检测等。

重点及难点是二叉判定图(BDD);符号模型检测算法;公平性。

15. 模型检测应用

主要包括:偏序约简、组合推理;模型检测 Procedural 程序;模型检测并发程序;模型检测安全协议;模型检测概率系统等。

重点及难点是各种模型检测算法及其复杂度问题。

七、考核要求

建议本课程结合研究生的课堂表现(回答问题、论文阅读、小组讨论等),对相关理论与工具的运用、课程论文、期末考试等几方面进行课程考核,形成最终考核成绩。各学科依托单位可根据实际情况,确定上述几方面在最终成绩中所占比例。

八、编写成员名单

本课程指南建议稿由谢冰(北京大学)主笔,并组织王捍贫(北京大学)、曹永知(北京大学)、郭耀(北京大学)等相关专家学者研讨撰写,李宣东(南京大学)审阅了建议稿,并反馈了修改意见和建议。特别感谢计算机科学与技术学科评议组的审议意见,以及在多次征求意见时收到的高校院所相关学科负责人及专家学者反馈的意见建议。

05 软件系统与工程

一、课程概述

推荐本课程作为计算机科学与技术一级学科的研究生核心课程。建议有关学科依托单位根据优势和特色,自行决定是否将本课程纳入培养方案的核心课程体系,自行决定是否需要调整本课程指南的建议内容。

软件系统与工程是计算机科学与技术的重要组成部分,并随着各种新型应用与系统的出现而快速发展,是研究生从事本学科相关领域研究与开发工作的重要基础。

本课程涉及基础软件系统、高级程序设计语言、软件工程等核心内容,为研究生提供较为全面的基础软件和工程技术知识,培养软件思维和工程实践能力。

通过本课程的学习,能够掌握并具备大型、复杂软件系统研究与开发的基础理论知识和工程实践能力,提升计算思维,了解新时代软件系统研究与开发的最新进展,为从事相关领域的研

研究与工程开发工作奠定坚实基础。

二、先修课程

操作系统,数据结构,编译原理,计算机程序设计,数据库,软件工程,计算机网络或分布式系统等。

三、课程目标

本课程应体现软件系统应用范围广、涉及面宽、随各种新型应用与系统的出现而快速发展等特点,建设成为一门系统性、实践性、前瞻性的研究生核心课程。

通过本课程的学习,能够掌握的知识、具备的能力主要体现在以下四个方面。

- (1) 掌握基础软件、高级程序设计语言和软件工程的一般技术和方法。
- (2) 了解新时代软件系统研究与开发的最新进展。
- (3) 理解云计算、大数据等对基础软件系统的要求与影响。
- (4) 掌握并具备大型、复杂软件系统开发和研究的基础理论知识和工程实践能力。

四、适用对象

计算机科学技术学科的硕士研究生和博士研究生,其他相关学科的研究生。

五、授课方式

建议本课程采用课堂讲授、实验实践、课堂研讨等相结合的教学方式。

(1) 课堂讲授。对于基础知识和发展趋势等内容,主要采用课堂讲授的方式进行教学。

(2) 实验实践。对于开发平台和技术,建议采用实验教学方式,在实践中使学生具备切实的实践能力。

(3) 课堂研讨。对于最新研究进展和深入学习需要,采用学生课外阅读相关文献、课堂研讨等方式进行教学。

因课程内容较广泛,建议本课程由多位教师授课,内容上应针对学生情况进行删减;建议以基础知识、原理作为知识点讲授,以实验教学提高学生动手能力。

建议加强从“软件定义”(SDx)的角度,分析软件技术本身的发展趋势和影响。

六、课程内容

建议本课程内容考虑涵盖基础软件系统、高级程序设计语言、软件工程等三部分。各学科依托单位可根据自身优势与特色,适当调整课程内容。

第一部分 基础软件系统

1. 基础软件系统概述

主要包括基础软件系统基本概念,系统软件和基础软件发展历史,主流软件工具/系统栈及其主要示例。

2. 操作系统进阶

主要包括操作系统基础知识回顾;虚拟化与云操作系统;实时操作系统,机器人操作系统的

发展趋势。

3. 数据库系统进阶

主要包括行存与列存;内存数据库管理系统;查询优化,查询的并发执行;数据密集型计算系统,如 SQL、NoSQL、NewSQL 概念;部分系统实例介绍。

4. 中间件与云计算

主要包括传统中间件系统及其架构,云计算基础设施,事务处理机制,区块链系统,边缘计算概念。

第二部分 高级程序设计语言

5. 程序设计语言发展、分类及其设计原理

6. 过程式程序设计语言、面向对象程序设计语言

7. 函数式程序设计语言及其实践

8. 逻辑程序设计语言及其实践

9. 并发程序设计语言与描述性程序设计语言

第三部分 软件工程部分

10. 软件工程概论

主要包括软件工程历史、软件开发范型、软件生存周期模型。

11. 软件项目管理

主要包括项目组织立项、估算与计划、执行与监控;软件工程度量和质量保障体系,如 GQM 和 GQM+度量体系、过程度量与产品度量、ISO9000 质量保证体系和 CMMI 过程改进。

12. 开源软件开发方法

开源协议简介,大教堂与市集、Linus 定律等规律,开源平台与工具简介。

13. 需求工程

主要包括软件需求描述,需求工程典型方法(包括面向目标的方法、面向主体的方法、面向情景的方法、问题驱动的方法等),非功能性需求分析技术,需求规约和验证技术,领域工程和基于领域建模的需求开发技术。

14. 软件体系结构

主要包括软件体系结构的内涵、典型软件体系结构风格;体系结构描述语言 ADL;软件设计模式;软件产品线工程;面向特定领域的软件体系结构介绍。

15. 程序分析与软件测试

主要包括软件质量模型、质量度量;软件测试的理论、方法和测试用例设计;程序静态分析的技术和工具;程序动态分析的技术和工具;软件测试技术和工具、自动化测试技术和框架;软件测试管理工具、缺陷跟踪系统介绍。

16. 模型驱动的软件开发方法

主要包括模型驱动导论,模型驱动工程技术架构,元模型理论、元建模标准、元-元模型体系、模型转换理论和模型转换标准等。

17. 敏捷软件开发方法

主要包括敏捷开发的原理和实践,以及敏捷开发过程的管理和协调等。

18. 软件复用与软件自动化

主要包括软件复用概念、历史和发展趋势,软件开发中的智能推荐技术,领域特定的软件自动化技术。

建议本课程针对一些具体内容设计实验环节,由研究生选择部分实验开展实践,培养动手实践能力。

七、考核要求

建议本课程考核成绩由期末考试、实验考核和课程大报告等三部分构成。各学科依托单位可根据实际情况,适当调整各部分比例。

(1) 书面考核(占总成绩 30%)。对基本概念、原理进行书面考查。

(2) 实验考核(占总成绩 40%)。按照所设置的课程实验的结果进行考核。

(3) 课程大报告(占总成绩 30%)。研究生自由选择感兴趣方向,进行论文研读和总结,形成技术综述报告。

八、编写成员名单

本课程指南建议稿由谢冰(北京大学)主笔,并组织孙艳春(北京大学)、胡振江(北京大学)、郭耀(北京大学)、金芝(北京大学)、黄罡(北京大学)等相关专家学者研讨撰写,李宣东(南京大学)审阅了建议稿,并反馈了修改意见和建议。特别感谢计算机科学与技术学科评议组的审议意见,以及在多次征求意见时收到的高校院所相关学科负责人及专家学者反馈的意见建议。

06 数据科学与工程

一、课程概述

推荐本课程作为计算机科学与技术一级学科的研究生核心课程。建议有关学科依托单位根据优势和特色,自行决定是否将本课程纳入培养方案的核心课程体系,自行决定是否调整本课程指南的建议内容。

当前,大数据已经成为一种新的国家战略资源,具有海量数据规模、快速数据流转、多样化数据类型和低价值密度等重要特征,在数据获取、存储、管理、分析及数据保护等方面催生了新的科学理论和工程方法,逐步形成了数据科学与大数据技术,并广泛应用到计算机科学与技术学科的众多领域。对数据科学与工程的深入理解是研究生从事数据驱动的科学研究和技术研发的基础,也是从事数据管理与分析处理相关研究与开发的重要基础。

本课程涉及数据科学与大数据概论、数据采集与预处理、数据管理、数据分析、数据可视化、数据安全与隐私保护、面向大数据处理的分布式计算平台与系统及典型数据处理方法等相关核心内容和最新研究进展,以及大数据典型应用领域的需求特征、数据特征与数据处理的特殊方