

编译原理混合学习教学实践探索

王挺

tingwang@nudt.edu.cn

国防科技大学计算机学院

2019.08.22 西宁

目录

- 编译原理课程的定位
- 教学实践情况
- 课程下一步考虑
- 总结

目录

- 编译原理课程的定位
- 教学实践情况
- 课程下一步考虑
- 总结

经常遇到的问题

- 为什么要学习编译原理?
- 不做编译器要不要学编译原理?

本科工程教育的定位

- 聚焦解决复杂工程问题的能力
 - 通过**基于原理的分析和设计**，解决复杂问题
 - 通过建立合适的**抽象模型**解决问题
 - 在理论指导下，进行工程技术实现
- 编译程序的构造是一个复杂工程问题
- 需要抽象、理论模型、系统设计与实现

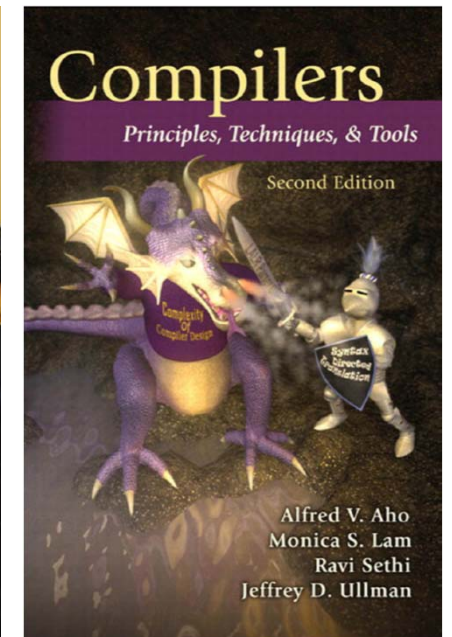
理论和**实践**相结合

关于编译课程

I have always enjoyed teaching the compilers course. Compiler design is a beautiful marriage of theory and practice -- it is one of the first major areas of systems programming for which a strong theoretical foundation has developed that is now routinely used in practice.



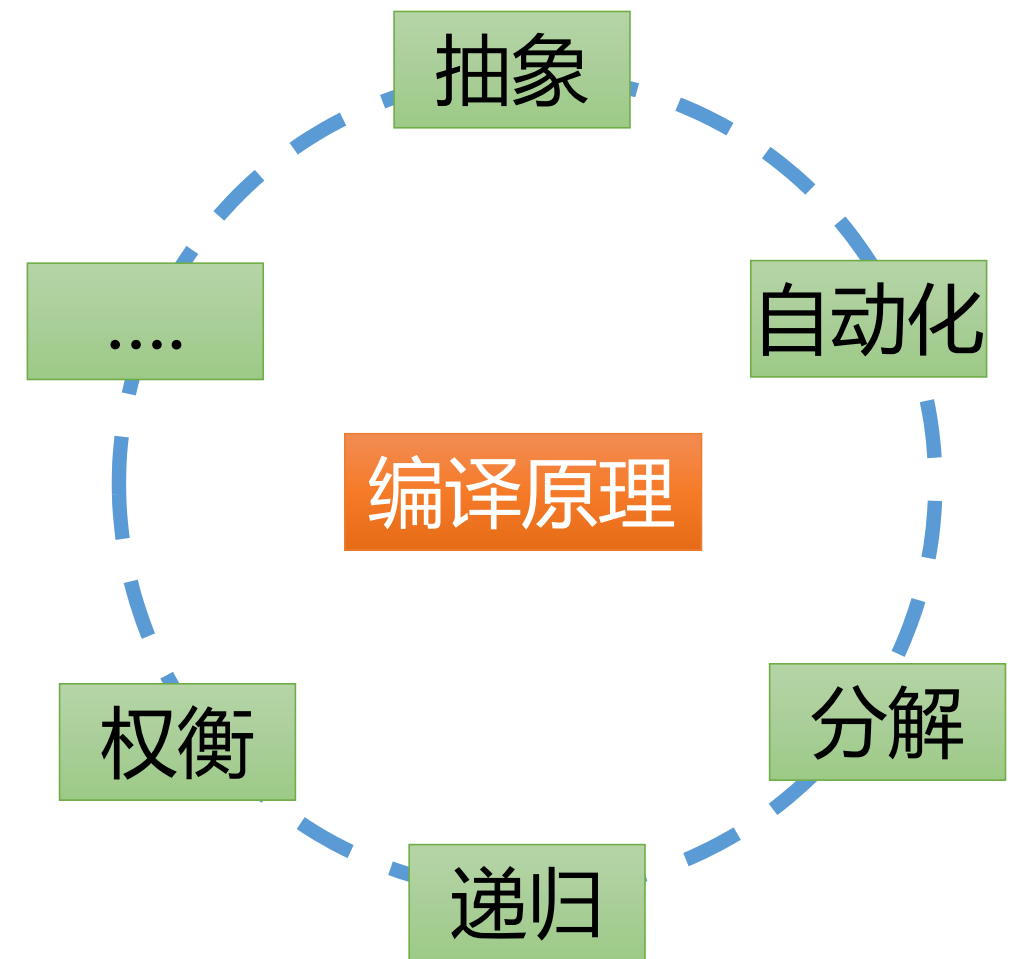
Alfred V. Aho



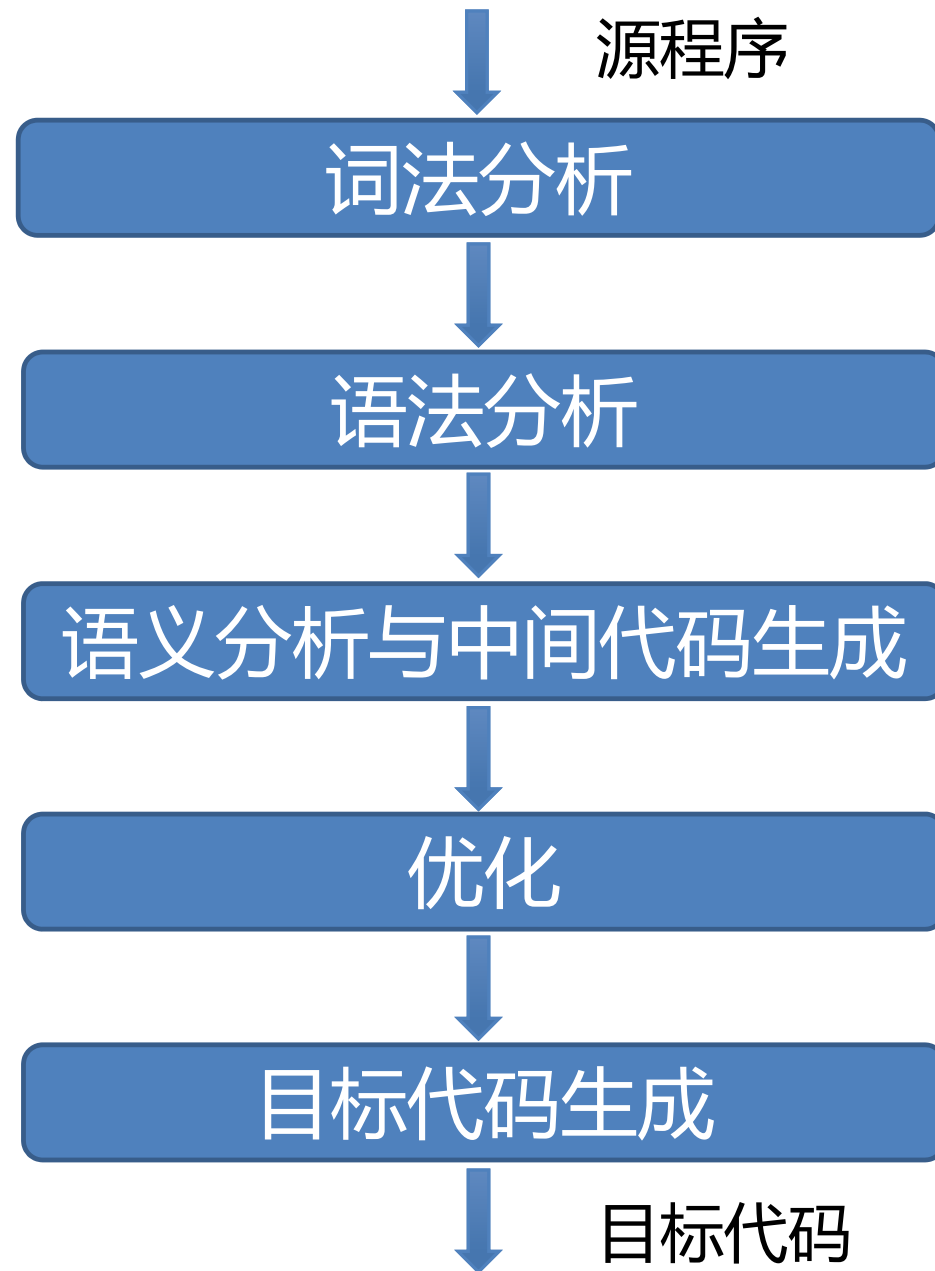
经常遇到的问题

- 为什么要学习编译原理?
- 不做编译器要不要学编译原理?

体验理论和实践的完美结合
学会通过抽象系统地解决(复杂工程)问题
欣赏计算思维的作用



编译课程



编译原理(本)

学时: 36

学分: 2

宽

以语言翻译的**基本**
原理为重点

学时: 36

学分: 2

高级编译技术(研)

深

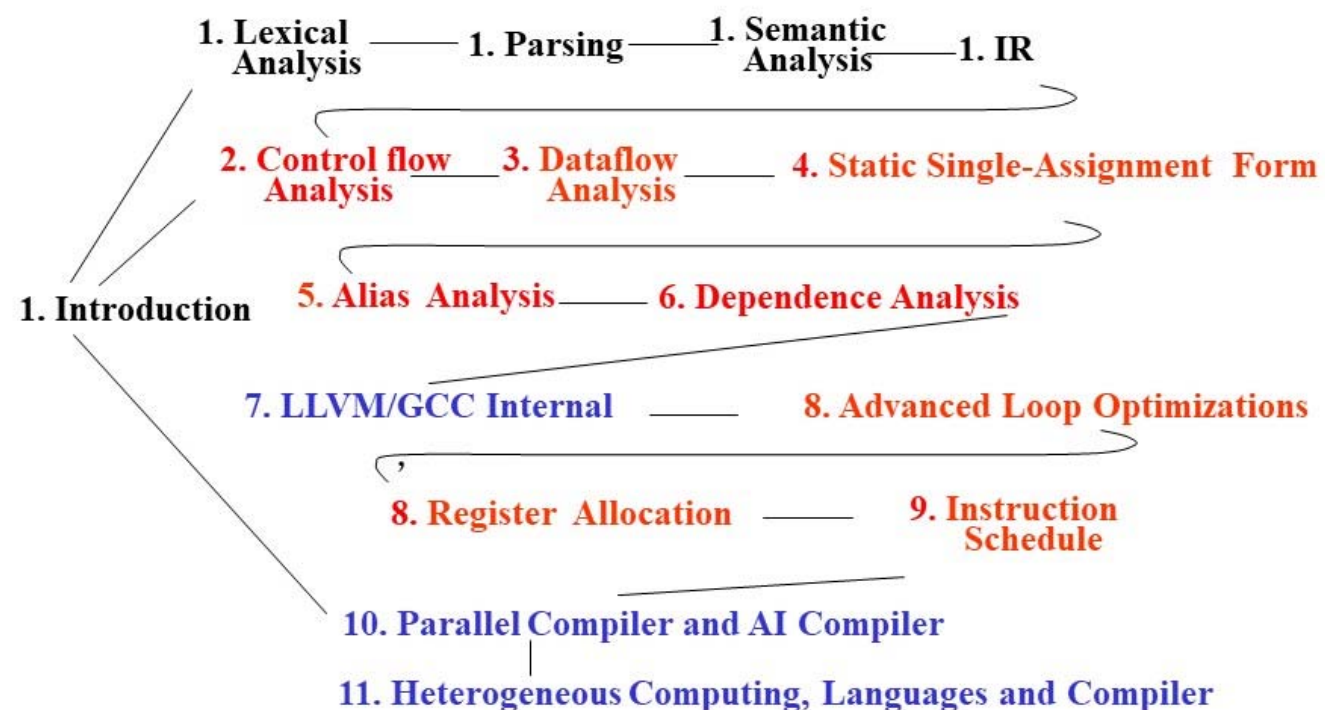
面向先进目标平
台的**前沿技术**



高级编译技术(研)

- 主讲：黄春 研究员

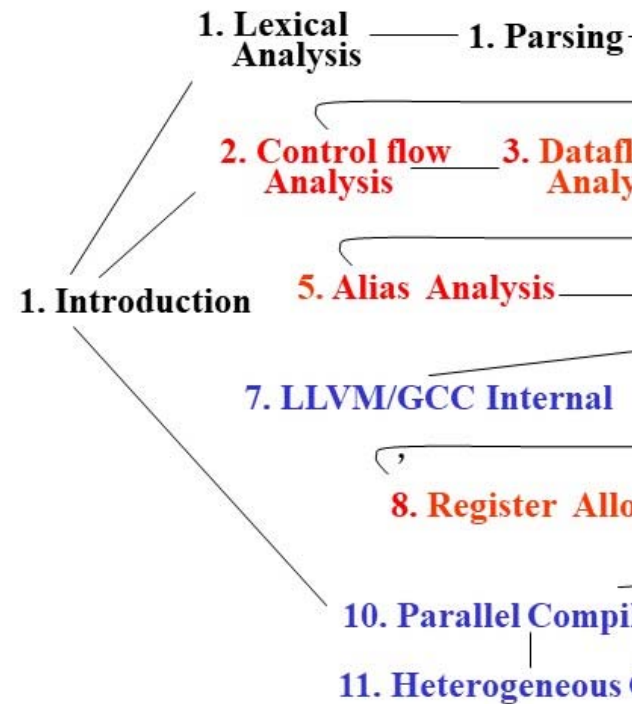
Course Syllabus



高级编译技术(研)

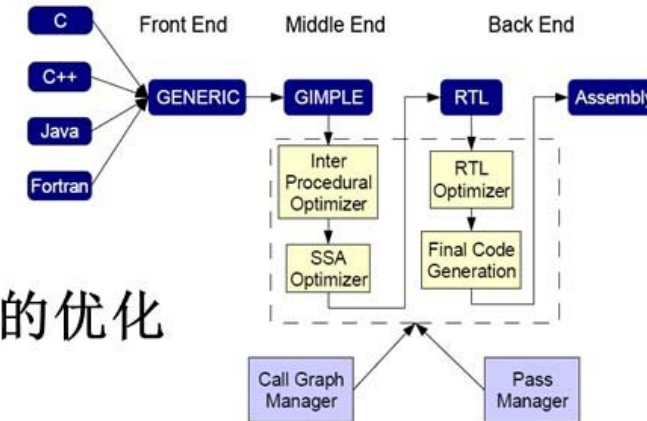
- 主讲：黄春 研究员

Course



Labs

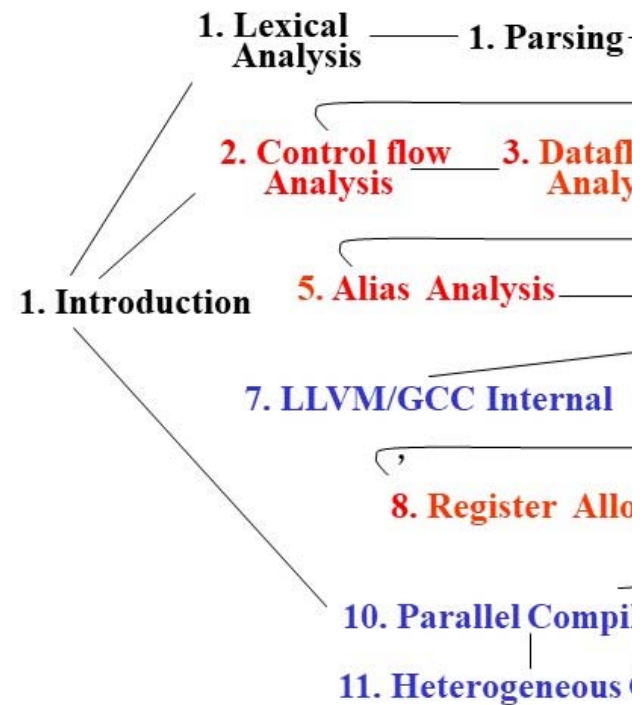
- **GCC framework**
 - ⊕ 多语言、多平台
 - ⊕ open source
 - ⊕ 应用广泛
 - ⊕ GIMPLE语法树一级的优化
- **目的**
 - ⊕ 了解GCC
 - ⊕ 理解和改善优化算法



高级编译技术(研)

- 主讲：黄春 研究员

Course



La

- GCC framework
 - ⊕ 多语言、多平台
 - ⊕ open source
 - ⊕ 应用广泛
 - ⊕ GIMPLE语法树一统
- 目的
 - ⊕ 了解GCC
 - ⊕ 理解和改善优化算法

Labs

- CUDA Compiler
 - ⊕ LLVM
 - ⊕ Open source
- 目的
 - ⊕ 理解GPU上程序的生成
 - ⊕ 实现新的语言



编译原理

计算思维

从语言翻译和表示变换的角度理解
计算和计算思维

计算机系统

深入理解程序语言和程序执行过程，
提高对**计算机系统**的总体认识。

系统软件

在理解编译系统的结构、工作流程以及编译程序各组成部分的设计原理和实现技术的基础上，获得分析、设计、实现和维护**编译系统**的初步能力。



Peter J. Denning

Computation is a sequence of representations.

目录

- 编译原理课程的定位
- 教学实践情况
- 课程下一步考虑
- 总结

编译原理的教学实践

- 利用经典理论成果吸引学生
- 通过应用案例拓展学生的视野
- 运用技术手段改进教学过程



目录

- 编译原理课程的定位
- 教学实践情况
 - 利用经典理论成果吸引学生
 - 通过应用案例拓展学生的视野
 - 运用技术手段改进教学过程
- 课程下一步考虑
- 总结

利用经典理论成果吸引学生

- 用图灵奖的成果激发兴趣

关于编译理论与技术

- ▶ 编译理论与技术
 - ▶ 计算机科学与技术中理论和实践相结合的最好典范
- ▶ ACM 图灵奖
 - ▶ 授予在计算机技术领域作出突出贡献的科学家
 - ▶ 程序设计语言、编译相关的获奖者是最多的

Analysis of Algorithms **Artificial Intelligence** Combinatorial Algorithms
Compilers
Computational Complexity Computer Architecture
Computer Hardware
Cryptography Data Structures Databases Education
Error Correcting Codes Finite Automata Graphics
Interactive Computing Internet Communications List Processing Numerical Analysis
Numerical Methods Object Oriented Programming Operating Systems Personal Computing
Program Verification Programming
Programming Languages Proof Construction Software
Theory Software Engineering
Verification of Hardware and Software Models Computer Systems Machine Learning
Parallel Computation

<http://amturing.acm.org/bysubject.cfm>



ACM图灵奖

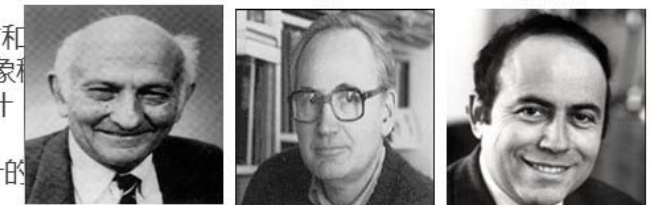


Alan J. Perlis Edsger W. Dijkstra John W. Backus Kenneth E. Iverson

- ▶ Alan J. Perlis (1966) -- ALGOL
- ▶ Edsger Wybe Dijkstra (1972) -- ALGOL
- ▶ Michael O. Rabin & Dana S. Scott (1976) --非确定自动机
- ▶ John W. Backus (1977) -- FORTRAN
- ▶ Kenneth Eugene Iverson (1979) -- APL程序语言
- ▶ Niklaus Wirth (1984) -- PASCAL
- ▶ John Cocke (1987) -- RISC & 编译优化
- ▶ O. Dahl, K.Nygaard (2001) -- Simula语言和
- ▶ Alan Kay(2003) -- SmallTalk语言和面向对象
- ▶ Peter Naur(2005) -- ALGOL60以及编译设计
- ▶ Frances E. Allen(2006)-- 优化编译器
- ▶ Barbara Liskov(2008)--编程语言和系统设计的
- ▶ ...



K.Nygaard O. Dahl



John Cocke Dana S. Scott Michael O. Rabin



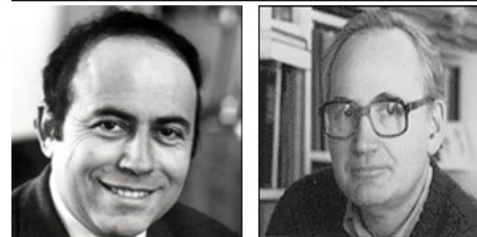
Niklaus Wirth Donald E. Knuth Barbara Liskov Frances E. Allen Peter Naur Alan Kay

利用经典理论成果吸引学生

- 用图灵奖的成果激发兴趣

非确定有限自动机

- ▶ 1976年图灵奖：For their joint paper "**Finite Automata and Their Decision Problem**," which introduced the idea of nondeterministic machines, which has proved to be an enormously valuable concept. Their (Scott & Rabin) classic paper has been a continuous source of inspiration for subsequent work in this field.



Michael O. Rabin Dana S. Scott

LR分析法

- ▶ 1965年 由Knuth提出
- ▶ L: 从左到右扫描输入串
- ▶ R: 自下而上进行归约



Donald Ervin Knuth

For his major contributions to the analysis of algorithms and the design of programming languages, and in particular for his contributions to **"the art of computer programming"** through his well-known books in a continuous series by this title.

利用经典理论成果吸引学生

- 用图灵奖的成果激发兴趣

图灵奖与编译优化



John Cocke
1987

JOHN COCKE

United States – 1987

CITATION

For significant contributions in the design and theory of compilers, the architecture of large systems and the development of reduced instruction set computers (RISC); for discovering and systematizing many fundamental transformations now used in optimizing compilers including reduction of operator strength, elimination of common subexpressions, register allocation, constant propagation, and dead code elimination.



Frances E. Allen
2006

FRANCES ("FRAN") ELIZABETH ALLEN

United States – 2006

CITATION

For pioneering contributions to the theory and practice of optimizing compiler techniques that laid the foundation for modern optimizing compilers and automatic parallel execution. [press release](#)

图灵奖与编译优化

- ▶ 用“图”表示程序的语句序列，以发现程序内在性质



John Cocke
1987



Frances E. Allen
2006

A CATALOGUE OF OPTIMIZING TRANSFORMATIONS

Frances E. Allen
John Cocke
IBM Thomas J. Watson Research Center
Yorktown Heights

Thomas J. Watson IBM Research Center. Research Division, F. E. Allen, and J. Cocke. *A catalogue of optimizing transformations.* 1971.

利用经典理论成果吸引学生

• 案例——DAG优化算法

理论模型

有向无环图(DAG)

▶ 有向图(Directed Graph)

- ▶ 有向边: $n_i \rightarrow n_j$
- ▶ 前驱: n_i 是 n_j 的前驱
- ▶ 后继: n_j 是 n_i 的后继
- ▶ 通路: $n_1 \rightarrow n_2, n_2 \rightarrow n_3, \dots, n_{k-1} \rightarrow n_k$
- ▶ 环路: $n_1 = n_k$

▶ 有向无环图(Directed Acyclic Graph, 简称 DAG)

- ▶ 没有环路

利用经典理论成果吸引学生

• 案例——DAG优化算法

模型应用

理论模型

DAG的扩充

有向无环图(DAG)

▶ 有向图(Directed Graph)

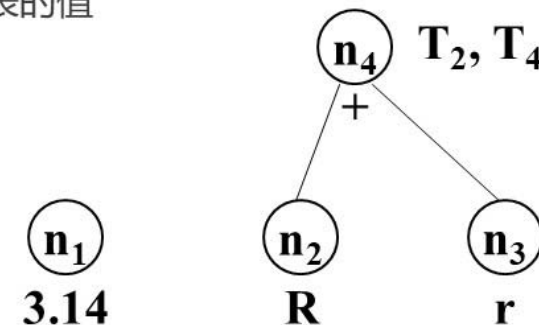
- ▶ 有向边: $n_i \rightarrow n_j$
- ▶ 前驱: n_i 是 n_j 的前驱
- ▶ 后继: n_j 是 n_i 的后继
- ▶ 通路: $n_1 \rightarrow n_2, n_2 \rightarrow n_3, \dots, n_{k-1} \rightarrow n_k$
- ▶ 环路: $n_1 = n_k$

▶ 有向无环图(Directed Acyclic Graph DAG)

- ▶ 没有环路

▶ 在DAG增加标记和附加信息

- ▶ 图的叶结点以一标识符或常数作为标记, 表示该结点代表该变量或常数的值
- ▶ 图的内部结点以一运算符作为标记, 表示该结点代表应用该运算符对其后继结点所代表的值进行运算的结果
- ▶ 各个结点上可能附加一个或多个标识符(称附加标识符)表示这些变量具有该结点所代表的值



34

利用经典理论成果吸引学生

• 案例——DAG优化算法

理论模型

模型应用

DAG的扩充

有向无环图(DAG)

- ▶ 有向图(Directed Graph)
 - ▶ 有向边: $n_i \rightarrow n_j$
 - ▶ 前驱: n_i 是 n_j 的前驱
 - ▶ 后继: n_j 是 n_i 的后继
 - ▶ 通路: $n_1 \rightarrow n_2, n_2 \rightarrow n_3, \dots, n_{k-1} \rightarrow n_k$
 - ▶ 环路: $n_1 = n_k$
- ▶ 有向无环图(Directed Acyclic Graph DAG)
 - ▶ 没有环路

- ▶ 在DAG增加标记和附加信息
 - ▶ 图的叶结点以一标识符或常数作为变量或常数的值
 - ▶ 图的内部结点以一运算符作为标记运算符对其后继结点所代表的值
 - ▶ 各个结点上可能附加一个或多个标识符, 这些变量具有该结点所代表的值

算法设计

1. 准备操作数的结点

- ▶ 如果NODE(B)无定义, 则构造一标记为B的叶结点并定义NODE(B)为这个结点;
 - ▶ 如果当前四元式是0型, 则记NODE(B)的值为n, 转4.
 - ▶ 如果当前四元式是1型, 则转2(1)
 - ▶ 如果当前四元式是2型, 则(i)如果NODE(C)无定义, 则构造一标记为C的叶结点并定义NODE(C)为这个结点; (ii)转2(2).

2. 合并已知量

- (1) 如果NODE(B)是标记为常数的叶结点, 则转2(3); 否则, 转3(1)
- (2) 如果NODE(B)和NODE(C)都是标记为常数的叶结点, 则转2(4); 否则, 转3(2)
- (3) 执行op B (即合并已知量)。令得到的新常数为P。如果NODE(B)是处理当前四元式时新构造出来的结点, 则删除它。如果NODE(P)无定义, 则构造一用P作标记的叶结点n。置NODE(P)=n, 转4
- (4) 执行B op C (即合并已知量)。令得到的新常数为P。如果NODE(B)或NODE(C)是处理当前四元式时新构造出来的结点, 则删除它。如果NODE(P)无定义, 则构造一用P作标记的叶结点n。置NODE(P)=n, 转4

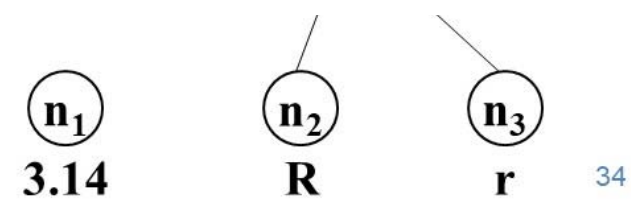
3. 删除公共子表达式

- (1) 检查DAG中是否已有一结点, 其唯一后继为NODE(B)且标记为op(即公共子表达式)。如果没有, 则构造该结点n, 否则, 把已有的结点作为它的结点并设该结点为n。转4
- (2) 检查DAG中是否已有一结点, 其左后继为NODE(B), 右后继为NODE(C), 且标记为op(即公共子表达式)。如果没有, 则构造该结点n, 否则, 把已有的结点作为它的结点并设该结点为n。转4。

4. 删除无用赋值

如果NODE(A)无定义, 则把A附加在结点n上并令NODE(A)=n; 否则, 先把A从NODE(A)结点上的附加标识符集中删除(注意, 如果NODE(A)是叶结点, 则其A标记不删除)。把A附加到新结点n上并置NODE(A)=n。转处理下一四元式。

A := op B
A := B op C



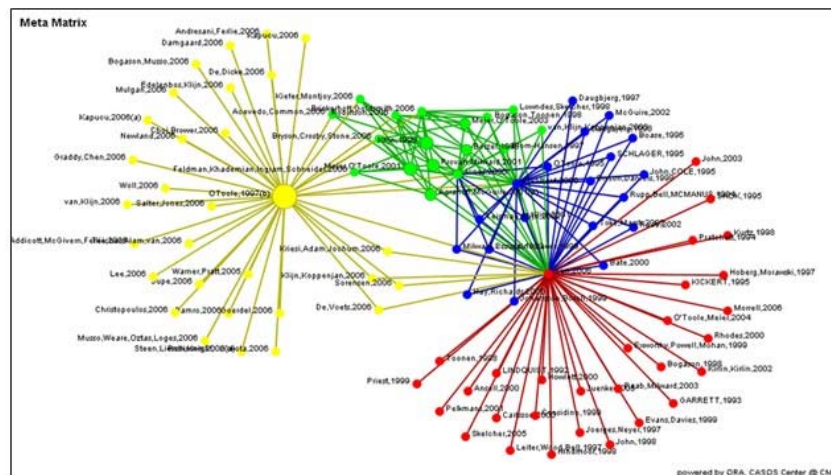
利用经典理论成果吸引学生

• 案例——DAG优化算法

新的应用

DAG的其他应用

- ▶ DAG拓扑结构独特，应用广泛
 - ▶ 动态规划，数据压缩，引文分析，社交网络.....



59

DAG的其他应用

- ▶ 区块链引入DAG，分布式账本主力竞争技术
 - ▶ 共识算法，去中心化机制.....



60



目录

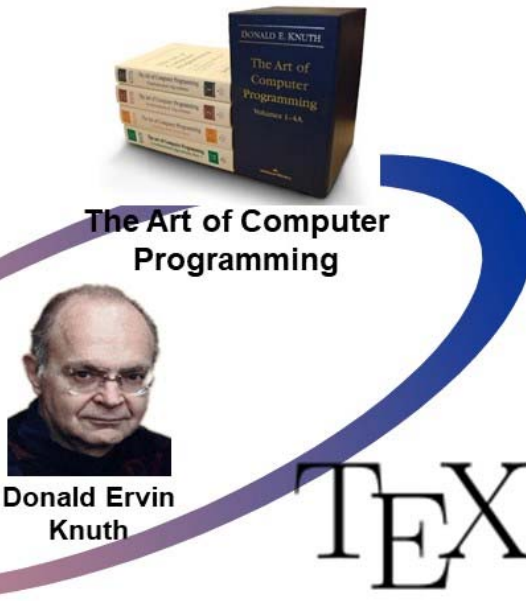
- 编译原理课程的定位
- 教学实践情况
 - 利用经典理论成果吸引学生
 - 通过应用案例拓展学生的视野
 - 运用技术手段改进教学过程
- 课程下一步考虑
- 总结

通过应用案例拓展学生的视野

- 理论与实践结合的案例

翻译模式示例

- ▶ 排版软件 TeX、LaTeX



The Art of Computer Programming

Donald Ervin Knuth

TeX

...
Lexical scanning
Parsing techniques
Context-Free Languages
Compiler Techniques
...

翻译模式示例

- ▶ 排版软件 TeX、LaTeX



Donald Ervin Knuth

The quadratic formula is $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

The quadratic formula is `\{-b\pm\sqrt{b^2-4ac} \over {2a}\}`

通过应用案例拓展学生的视野

- 理论与实践结合的案例

翻译模式示例

- ▶ 排版软件 TeX、LaTeX



Donald Ervin Knuth

The quadratic formula is $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

The quadratic formula is `\{-b\pm\sqrt{b^2-4ac} \over {2a}\}`

通过应用案例拓展学生的视野

- 理论与实践结合的案例

翻译模式示例

数学格式语言EQN

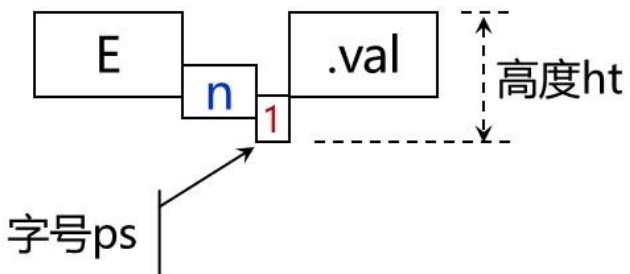
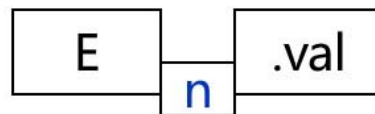
▶ 排版软件 TeX,

▶ 给定输入

E sub n .val

E sub n sub 1.val

The quadratic



The quadratic form

通过应用案例拓展学生的视野

- 理论与实践结合的案例

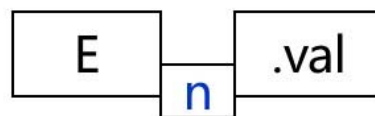
翻译模式示例

▶ 排版软件 TeX,

数学格式语言I

▶ 给定输入

E sub n .val



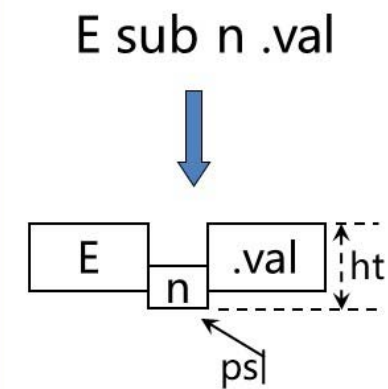
The quadratic

The quadratic form

数学格式语言EQN

▶ 识别输入并进行格式转换的L-属性文法

产生式	语义规则
$S \rightarrow B$	$B.ps := 10$ $S.ht := B.ht$
$B \rightarrow B_1 B_2$	$B_1.ps := B.ps$ $B_2.ps := B.ps$ $B.ht := \max(B_1.ht, B_2.ht)$
$B \rightarrow B_1 \text{ sub } B_2$	$B_1.ps := B.ps$ $B_2.ps := \text{shrink}(B.ps)$ $B.ht := \text{disp}(B_1.ht, B_2.ht)$
$B \rightarrow \text{text}$	$B.ht := \text{text.h} \times B.ps$



通过应用案例拓展学生的视野

• 理论与实践结合的案例

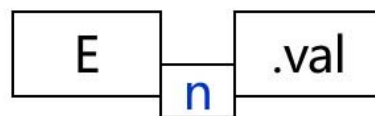
翻译模式示例

▶ 排版软件 TeX、

数学格式语言I

▶ 给定输入

E sub n .val



The quadratic

The quadratic form

数学格式语言EQN

▶ 识别输入并进行格式转

产生式	语义规则
$S \rightarrow B$	$B.ps := 10$ $S.ht := B.ht$
$B \rightarrow B_1 B_2$	$B_1.ps := B.ps$ $B_2.ps := B.ps$ $B.ht := \max(B_1.ht, B_2.ht)$
$B \rightarrow B_1 \text{ sub } B_2$	$B_1.ps := B.ps$ $B_2.ps := \text{shrink}(B.ps)$ $B.ht := \text{disp}(B_1.ht, B_2.ht)$
$B \rightarrow \text{text}$	$B.ht := \text{text.h}$

翻译模式

$S \rightarrow \{ B.ps:=10 \} B$
$\{ S.ht:=B.ht \}$
$B \rightarrow \{ B_1.ps:=B.ps \} B_1$
$\{ B_2.ps:=B.ps \} B_2$
$\{ B.ht:=\max(B_1.ht, B_2.ht) \}$
$B \rightarrow \{ B_1.ps:=B.ps \} B_1$
sub
$\{ B_2.ps:=\text{shrink}(B.ps) \} B_2$
$\{ B.ht:=\text{disp}(B_1.ht, B_2.ht) \}$
$B \rightarrow \text{text} \{ B.ht:=\text{text.h} \times B.ps \}$

产生式	语义规则
$S \rightarrow B$	$B.ps := 10$ $S.ht := B.ht$
$B \rightarrow B_1 B_2$	$B_1.ps := B.ps$ $B_2.ps := B.ps$ $B.ht := \max(B_1.ht, B_2.ht)$
$B \rightarrow B_1 \text{ sub } B_2$	$B_1.ps := B.ps$ $B_2.ps := \text{shrink}(B.ps)$ $B.ht := \text{disp}(B_1.ht, B_2.ht)$
$B \rightarrow \text{text}$	$B.ht := \text{text.h} \times B.ps$

1. 产生式右边符号的继承属性必须在该符号以前的动作中计算出来
2. 一个动作不能引用该动作右边符号综合属性
3. 产生式左边非终结符的综合属性只有在它所引用的所有属性都计算出来后才能计算



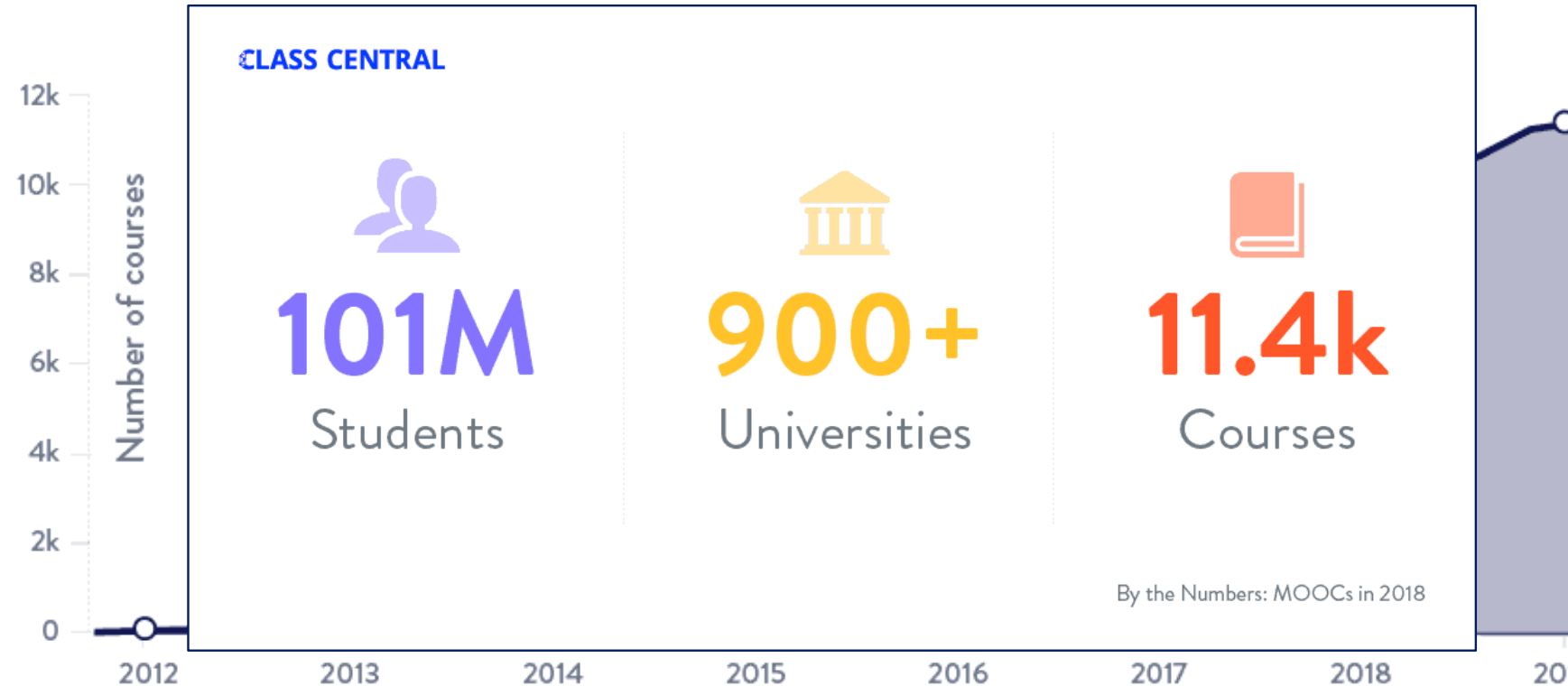
目录

- 编译原理课程的定位
- 教学实践情况
 - 利用经典理论成果吸引学生
 - 通过应用案例拓展学生的视野
 - 运用技术手段改进教学过程
- 课程下一步考虑
- 总结

在线学习的发展

- 大规模在线开放课程(MOOC)和其他模式在线教育

CLASS CENTRAL



Computer Science

1161 courses

Artificial Intelligence Algorithms and Data Structures
 Internet of Things Information Technology
 Cybersecurity Computer Networking
 Machine Learning DevOps Deep Learning
 Blockchain and Cryptocurrency

Data Science

480 courses

Bioinformatics Big Data Data Mining Data Analysis
 Data Visualization

Programming

916 courses

Mobile Development Web Development Databases
 Android Development iOS Development
 Game Development Programming Languages
 Software Development

By the Numbers: MOOCs in 2018

<https://www.classcentral.com/report/moocs-stats-and-trends-2018/>

线上线下结合改进教学过程

- 在线学习平台
 - 数据收集方便
 - 学习过程跟踪与控制
 - 学习效果评估
- 利用在线实训平台



在线课程的实践

中国大学MOOC 课程 名校 2020考研 名师专栏 客户端 搜索感兴趣的课程 我的学校云

首页 > 全部课程 > 计算机



编译原理

分享   

第2次开课 ▾

开课时间: 2019年03月04日 ~ 2019年07月01日

当前开课已结束

学时安排: 3-5小时每周

已有8793人参加

已参加, 查看内容

恭喜你已提前报名下一次开课, 开课信息会第一时间提醒你哦~ [取消报名](#)



<https://www.icourse163.org/course/NUDT-1003101005>

第1讲 引论

- 1.1 什么是编译程序
- 1.2 为什么要学习编译原理
- 1.3 编译过程
- 1.4 编译程序的结构
- 1.5 编译程序的生成
- 1.6 小结

第2讲 高级程序设计语言概述

- 2.1 常用的高级程序设计语言
- 2.2 程序设计语言的定义
- 2.3 高级程序设计语言的一般特性
- 2.4 小结

第3讲 高级程序设计语言的语法描述

- 3.1 上下文无关文法
- 3.2 文法与语言
- 3.3 语法树与二义性
- 3.4 形式语言鸟瞰
- 3.5 小结

第4讲 词法分析1

- 4.1 词法分析概述
- 4.2 词法分析器的设计
- 4.3 小结

第5讲 词法分析2

- 5.1 词法规则形式化——正规集与正规式
- 5.2 确定有限自动机
- 5.3 非确定有限自动机
- 5.4 小结

第6讲 词法分析3

- 6.1 有限自动机的等价性
- 6.2 正规式与有限自动机的等价性
- 6.3 词法分析程序自动生成

第1次单元测试

第7讲 语法分析——自上而下分析1

- 7.1 自上而下分析的基本问题
- 7.2 LL(1)文法——消除文法的左递归
- 7.3 LL(1)文法——消除回溯
- 7.4 FIRST和FOLLOW集合的构造
- 7.5 小结

第8讲 语法分析——自上而下分析2

- 8.1 构造递归下降分析器
- 8.2 扩充的巴科斯范式和语法图
- 8.3 JavaCC简介
- 8.4 小结

第9讲 语法分析——自上而下分析3

- 9.1 预测分析程序
- 9.2 预测分析表的构造
- 9.3 小结

第10讲 语法分析——自下而上分析1

- 10.1 自下而上分析
- 10.2 短语与直接短语
- 10.3 分析过程描述
- 10.4 算符优先文法
- 10.5 构造优先关系表
- 10.6 算符优先分析算法
- 10.7 小结

第11讲 语法分析——自下而上分析2

- 11.1 句柄和规范归约
- 11.2 LR分析法
- 11.3 小结

第12讲 语法分析——自下而上分析3

- 12.1 活前缀
- 12.2 构造识别活前缀的DFA
- 12.3 构造LR(0)分析表
- 12.4 小结

第13讲 语法分析——自下而上分析4

- 13.1 SLR(1)分析法
- 13.2 LR(1)分析法
- 13.3 LR分析器产生工具
- 13.4 小结

第2次单元测试

第14讲 属性文法和语法制导翻译1

- 14.1 属性文法
- 14.2 属性计算

第15讲 属性文法和语法制导翻译2

- 15.1 S-属性文法
- 15.2 L-属性文法
- 15.3 翻译模式
- 15.4 递归下降翻译器的设计
- 15.5 小结

第16讲 语义分析和中间代码生成1

- 16.1 中间语言
- 16.2 常用的中间语言形式
- 16.3 小结

第3次单元测试

第17讲 语义分析和中间代码生成2

- 17.1 赋值语句的翻译
- 17.2 数组元素引用的翻译
- 17.3 类型转换
- 17.4 小结

第18讲 语义分析和中间代码生成3

- 18.1 布尔表达式及其计算
- 18.2 按数值表示法翻译布尔表达式
- 18.3 带优化翻译布尔表达式
- 18.4 小结

第19讲 语义分析和中间代码生成4

- 19.1 常用的控制语句
- 19.2 控制语句的属性文法
- 19.3 控制语句的属性计算
- 19.4 一遍扫描翻译控制语句
- 19.5 一遍扫描翻译控制语句示例
- 19.6 小结

第4次单元测试

第20讲 符号表

- 20.1 符号表的组织与操作
- 20.2 符号表的内容
- 20.3 利用符号表分析名字的作用域
- 20.4 小结

第21讲 运行时存储空间组织1

- 21.1 参数传递
- 21.2 目标程序运行时的活动
- 21.3 静态存储管理

第22讲 运行时存储空间组织2

- 22.1 动态存储管理概述
- 22.2 非嵌套过程语言的动态存储管理
- 22.3 静态链方法
- 22.4 Display表方法
- 22.5 小结

第23讲 优化1

- 23.1 优化概述
- 23.2 局部优化--基本块划分
- 23.3 局部优化--基本块优化

第24讲 优化2

- 24.1 循环优化概述
- 24.2 代码外提
- 24.3 强度削弱
- 24.4 小结

第25讲 代码生成

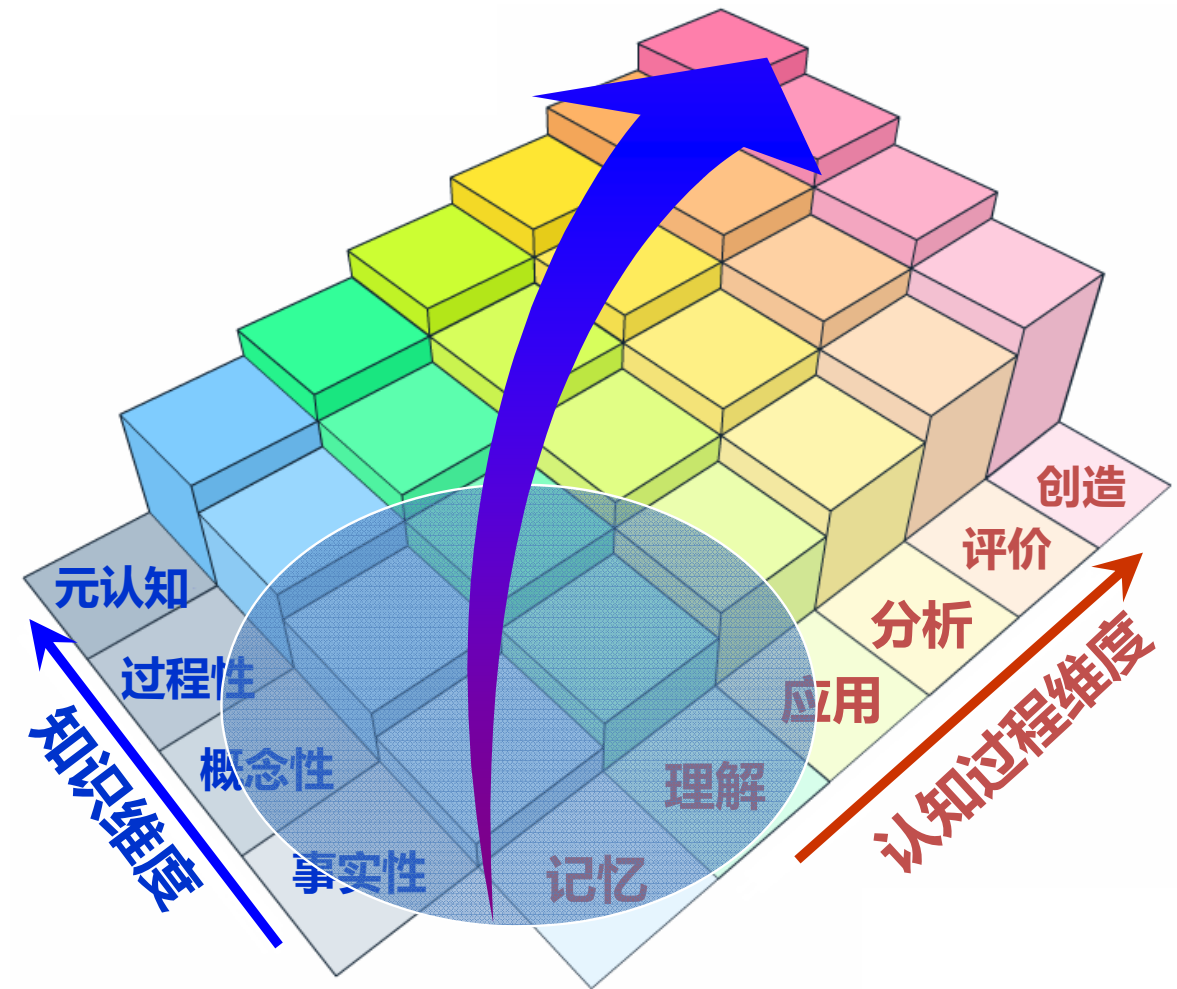
- 25.1 目标代码生成概述
- 25.2 目标机器模型
- 25.3 简单代码生成器
- 25.4 待用信息和活跃信息
- 25.5 变量地址描述和寄存器描述
- 25.6 代码生成与寄存器分配算法
- 25.7 小结

第5次单元测试

结课考试

利用MOOC开展混合学习

- 开设SPOC
- 改进教学过程



布鲁姆教育目标分类法

利用MOOC开展混合学习

- 课外学习的安排
 - 课前/课后线上自学



编译原理
王挺、刘春林、周会平、黄春、唐晋韬

课程 > 第4讲 词法分析1 > 4.2 词法分析器的设计

公告

评分标准

课件

测验与作业

考试

讨论区

课程分享

微信提醒课程进度

中国大学MOOC

状态转换图

- ▶ 状态转换图可用于识别(或接受)一定的字符串
 - ▶ 若存在一条从初态到某一终态的道路,且这条路上所有弧上的标记符连接成的字等于 α ,则称 α 被该状态转换图所识别(接受)

数字

```

    graph LR
        S(( )) --> 1((1))
        1 -- 数字 --> 2((2))
        2 -- 数字 --> 2
        2 -- 其他 --> 3(((3)))
        style S fill:none,stroke:none
    
```

识别整常数的状态转换图

若干个数字字符构成的串

利用MOOC开展混合学习

- 课外学习的安排
 - 课前/课后线上自学
- 关注学习进度和效果
 - 通过线上学习记录了解进度

整体学习人数

视频观看人数

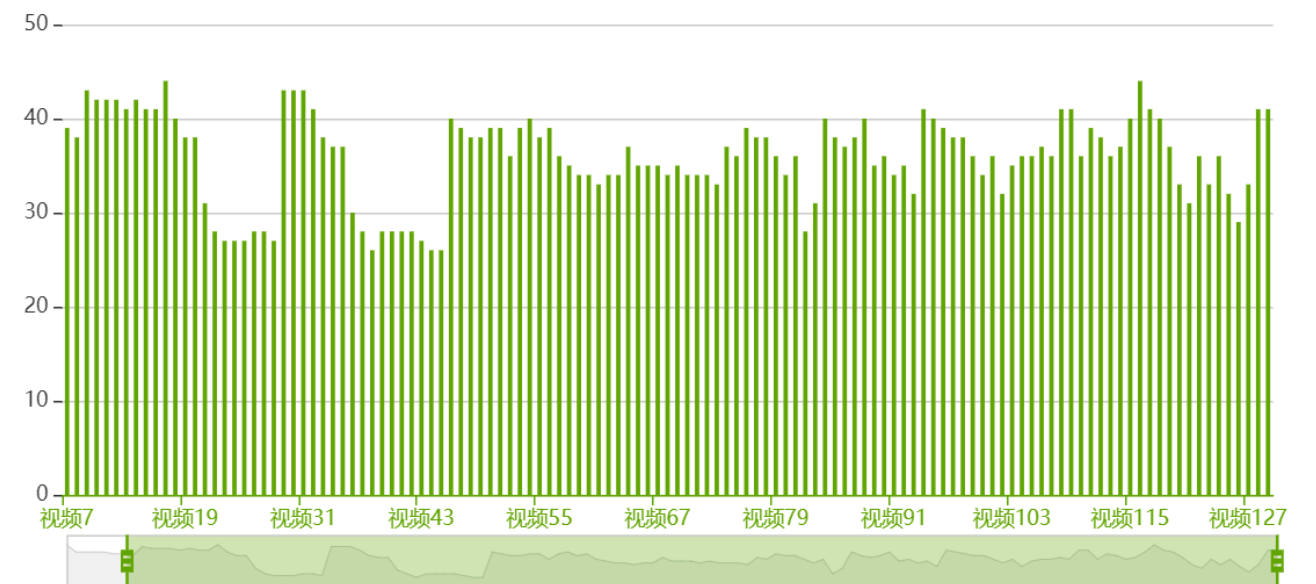
文档浏览人数

富文本浏览人数

随堂测验参与人数

随堂讨论参与人数

单元测验、单元作业和考试



利用MOOC开展混合学习

- 课外学习的安排
 - 课前/课后线上自学
- 关注学习进度和效果
 - 通过线上学习记录了解进度
 - 通过课堂调查检查学习效果

测试：优化的概念

- ▶ 下面关于优化的说法中，错误的是（ ）
- A. 优化遵循三个原则：等价原则，有效原则，合算原则
 - B. 优化分为三个级别：局部优化，循环优化和全局优化
 - C. 优化通常进行多遍扫描
 - D. 优化措施和目标平台无关



选项	小计	比例
优化遵循三个原则：等价原则，有效原则，合算原则	0	0%
优化分为三个级别：局部优化，循环优化和全局优化	1	2.27%
优化通常进行多遍扫描	4	9.09%
优化措施和目标平台无关	39	88.64%
本题有效填写人次	44	

利用MOOC开展混合学习

- 课内教学的组织
 - 利用在线平台获得实时反馈
 - 组织研讨
 - 评估学习效果

选项	小计	比例
有问题	17	45.95%
没有问题	20	54.05%
本题有效填写人次	37	

E'不考虑Follow集合有问题吗？

- A. 有问题
- B. 没有问题

▶ 文法G(E):

$E \rightarrow TE'$
 $E' \rightarrow +TE' \mid \epsilon$
 $T \rightarrow FT'$
 $T' \rightarrow *FT' \mid \epsilon$
 $F \rightarrow (E) \mid i$

▶ 对应的递归

```

PROCEDURE
BEGIN
    T; E'
END;
    
```



```

PROCEDURE E';
IF SYM= '+' THEN
BEGIN
    ADVANCE;
    T; E'
    
```

LOW(E')={), # }

```

RE E';
THEN
    
```

E;

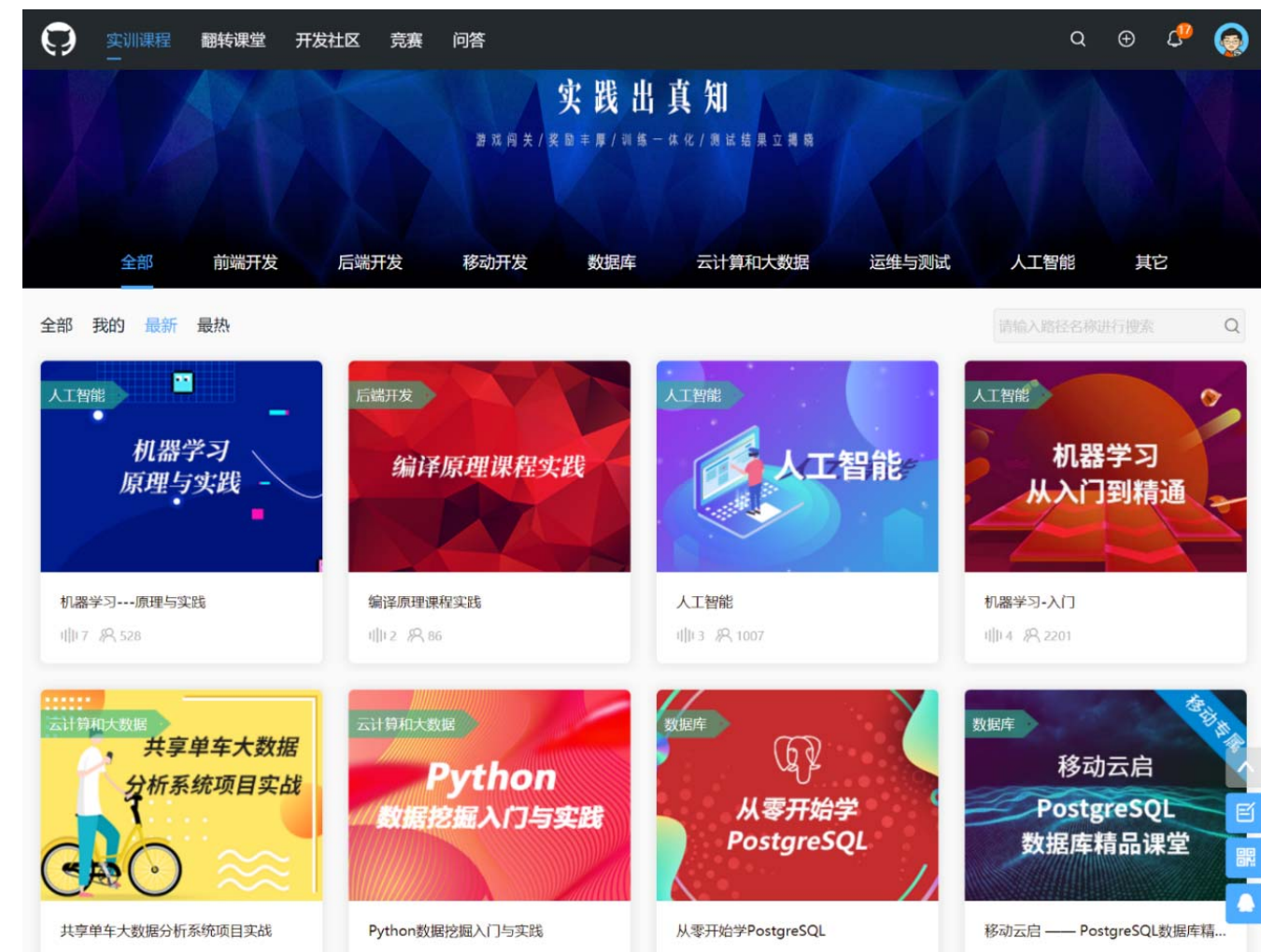
```

END
ELSE IF SYM<>'#' AND SYM<>')'
THEN ERROR
    
```

利用在线实训平台在开展课程实验

• 问题

- 如何定义任务、目标
- 如何进行过程控制
- 如何检验完成情况



www.educoder.net

利用在线实训平台在开展课程实验

• 编译课程实验

- 通过动手实践，使学生理解编译的基本过程、各个编译阶段的功能，能够利用LEX(FLEX)、JavaCC和YACC(BISON)等经典工具，设计并实现给定语言的词法分析、语法分析、语义分析等功能。
- 通过分析和修改语言的编译程序和解释程序，获得综合利用编译程序设计的知识进行分析、设计、实现和维护编译程序的能力。
- 通过课程实验，提高利用理论知识和自动化工具解决复杂问题的能力，加强从语言翻译和表示变换的角度对计算的理解



<https://www.educoder.net/paths/190>

利用在线实训平台在开展课程实验

- 编译原理课程实验
 - 单元实验：词法分析、语法分析和翻译
 - 用LEX(FLEX)生成PL语言的词法分析器
 - 用JAVACC生成并扩充C语言的语法分析器
 - 用YACC(BISON)生成语法分析和翻译器
 - 综合实验：典型编译程序扩充
 - 典型语言编译程序扩充
 - 综合实验：自主设计语言的翻译
 - 乐谱描述语言



编译原理课程实践 发送至 学习统计

章节	实训	实践任务	经验值	学习人数
2	4	9	4500	179

简介

编译原理是一门理论性、技术性、实践性很强的课程，理论和实践相结合是课程的重要特色。本实验课程旨在通过一组实训攻关，达到下列目标：

1. 通过动手实践，使学生理解编译的基本过程运用各个编译阶段的功能，能够利用LEX(FLEX)、JavaCC和YACC(BISON)等经典工具，设计并实现给定语言的词法分析、语法分析、语义分析等功能。
2. 通过分析和修改PL语言的编译程序和解释程序，获得综合利用编译程序设计的知识进行分析、设计、实现和维护编译程序的能力。
3. 通过课程实验，提高利用理论和自动化工具解决复杂问题的能力，加强从语言翻译和表示变换的角度对计算的理解。

课程须知

本课程安排了4个实训作业，其中3个是完成编译工具的单元实验，包括：

- 1.用LEX(FLEX)生成PL语言的词法分析器
- 2.用JAVACC生成并扩充C语言的语法分析器
- 3.YACC(BISON)工具的使用

第4个实训“PL语言编译器”是综合实验，要求同学们分析PL语言的编译程序和解释程序，然后实现对PL语言的扩充。

技能标签 7

入门 实践能手 自虐狂
超级自虐狂 熟练 进阶
编译老鸟

我的进展 0 / 4500

已学 0% 耗时: 0天 0小时 0分钟 0秒

教学团队

王挺 教授 绿盟杯六赛



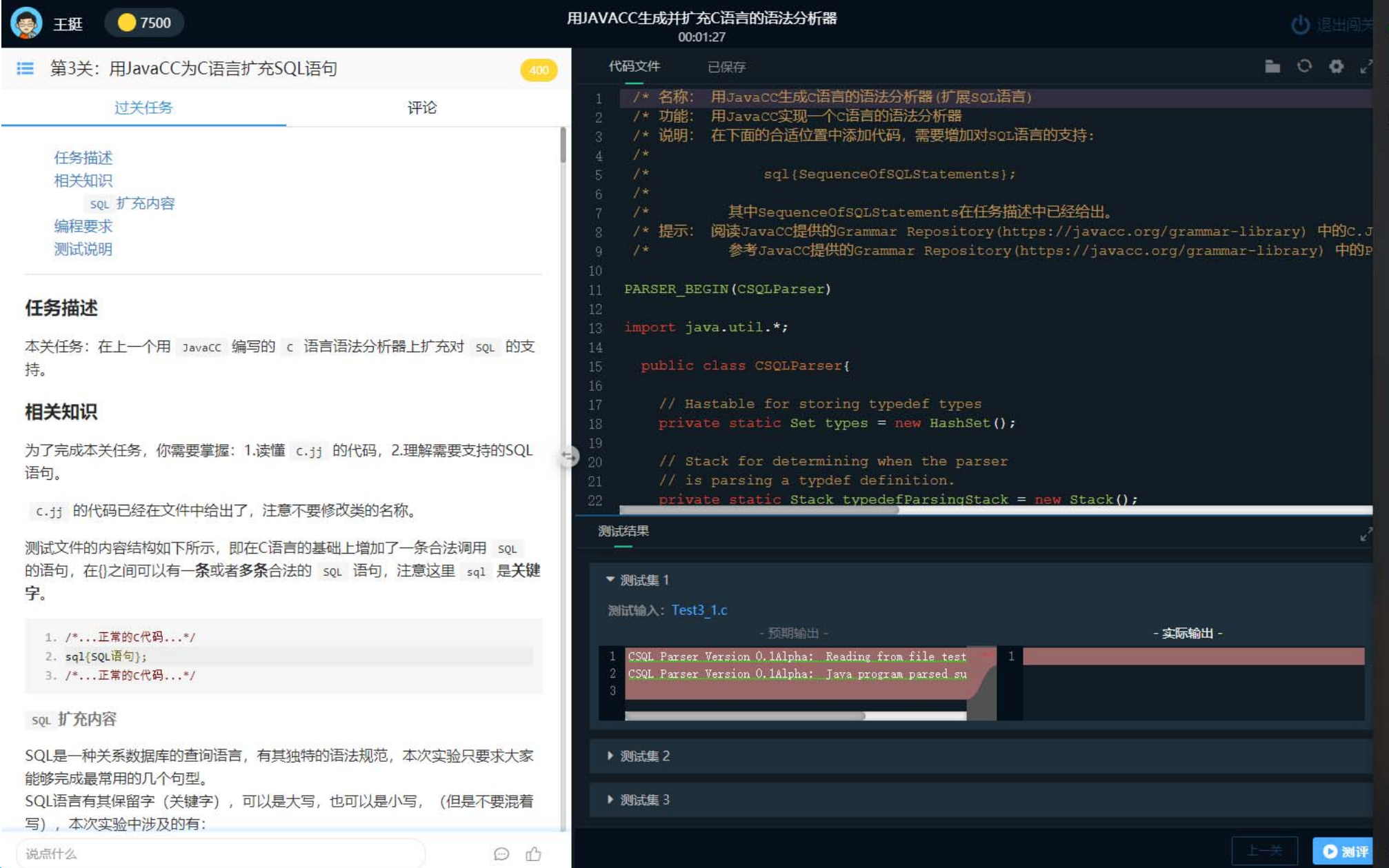
利用在线实训平台在开展课程实验

- 自主设计一个小的语言并进行翻译
 - 2-3人一组
 - 第5周前给出项目的建议书(提供模板), 通过后
 - 第8周前给出语言和翻译的定义
 - 第15周前完成系统设计与实现, 验收

利用在线实训平台在开展课程实验

- 测试驱动
- 逐级攻关
- 自动评估

- 第1关：熟悉JavaCC
- 第2关：构建C语言的语法分析器
- 第3关：用JavaCC为C语言扩充SQL语句



The screenshot shows a task page titled "第3关：用JavaCC为C语言扩充SQL语句" (Level 3: Expanding SQL statements for C language using JavaCC). The page includes a task description, related knowledge, and a code editor.

任务描述
本关任务：在上一个用 JavaCC 编写的 c 语言语法分析器上扩充对 sql 的支持。

相关知识
为了完成本关任务，你需要掌握：1.读懂 c.jj 的代码，2.理解需要支持的SQL 语句。
c.jj 的代码已经在文件中给出了，注意不要修改类的名称。

测试文件的内容结构如下所示，即在C语言的基础上增加了一条合法调用 sql 的语句，在{}之间可以有一条或者多条合法的 sql 语句，注意这里 sql 是关键字。

```

1. /*...正常的c代码...*/
2. sql{SQL语句};
3. /*...正常的c代码...*/
    
```

SQL 扩充内容
SQL是一种关系数据库的查询语言，有其独特的语法规则，本次实验只要求大家能够完成最常用的几个句型。
SQL语言有其保留字（关键字），可以是大写，也可以是小写，（但是不要混着写），本次实验中涉及的有：

代码文件 (已保存)

```

1  /* 名称： 用JavaCC生成c语言的语法分析器 (扩展SQL语言)
2  /* 功能： 用JavaCC实现一个c语言的语法分析器
3  /* 说明： 在下面的合适位置中添加代码，需要增加对SQL语言的支持：
4  /*
5  /*          sql{SequenceOfSQLStatements};
6  /*
7  /*          其中SequenceOfSQLStatements在任务描述中已经给出。
8  /* 提示： 阅读JavaCC提供的Grammar Repository (https://javacc.org/grammar-library) 中的C.J
9  /*          参考JavaCC提供的Grammar Repository (https://javacc.org/grammar-library) 中的p
10
11  PARSER_BEGIN(CSQLParser)
12
13  import java.util.*;
14
15  public class CSQLParser{
16
17      // Hashtable for storing typedef types
18      private static Set types = new HashSet();
19
20      // Stack for determining when the parser
21      // is parsing a typedef definition.
22      private static Stack typedefParsingStack = new Stack();
    
```

测试结果

测试集 1
测试输入: Test3_1.c

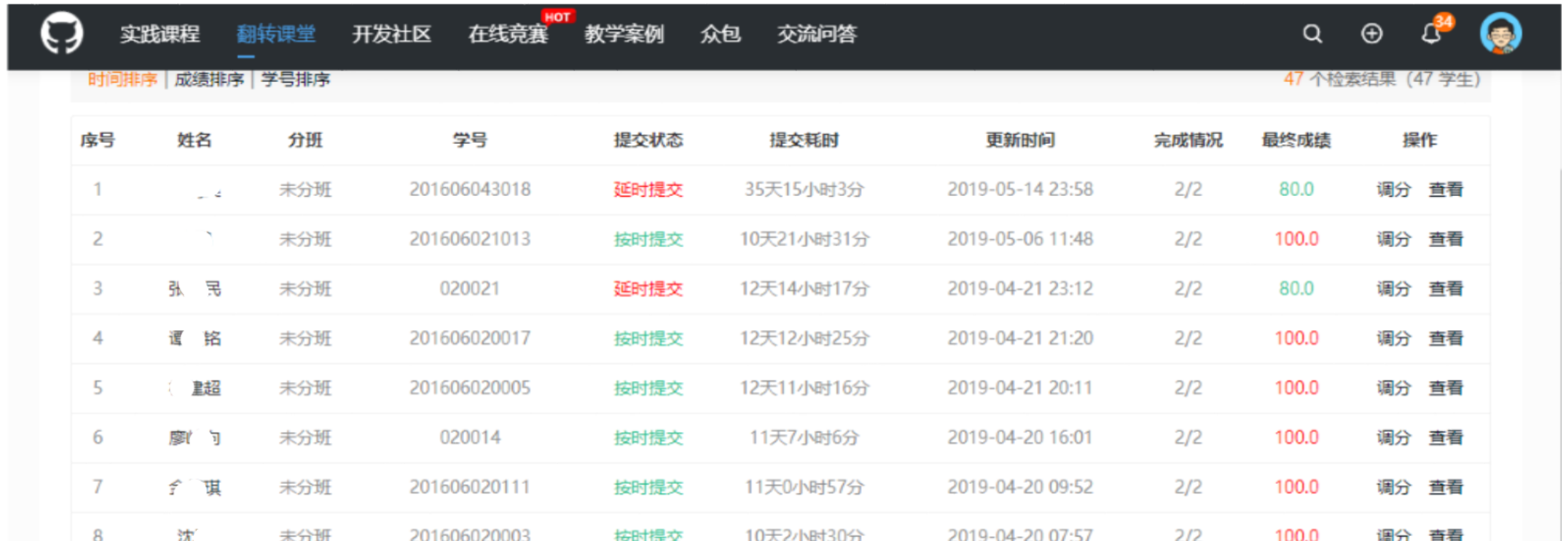
预期输出	实际输出
1 CSQL Parser Version 0.1Alpha: Reading from file test	1
2 CSQL Parser Version 0.1Alpha: Java program parsed su	
3	

测试集 2
测试集 3

说点什么

利用在线实训平台在开展课程实验

- 根据过关情况自动评分

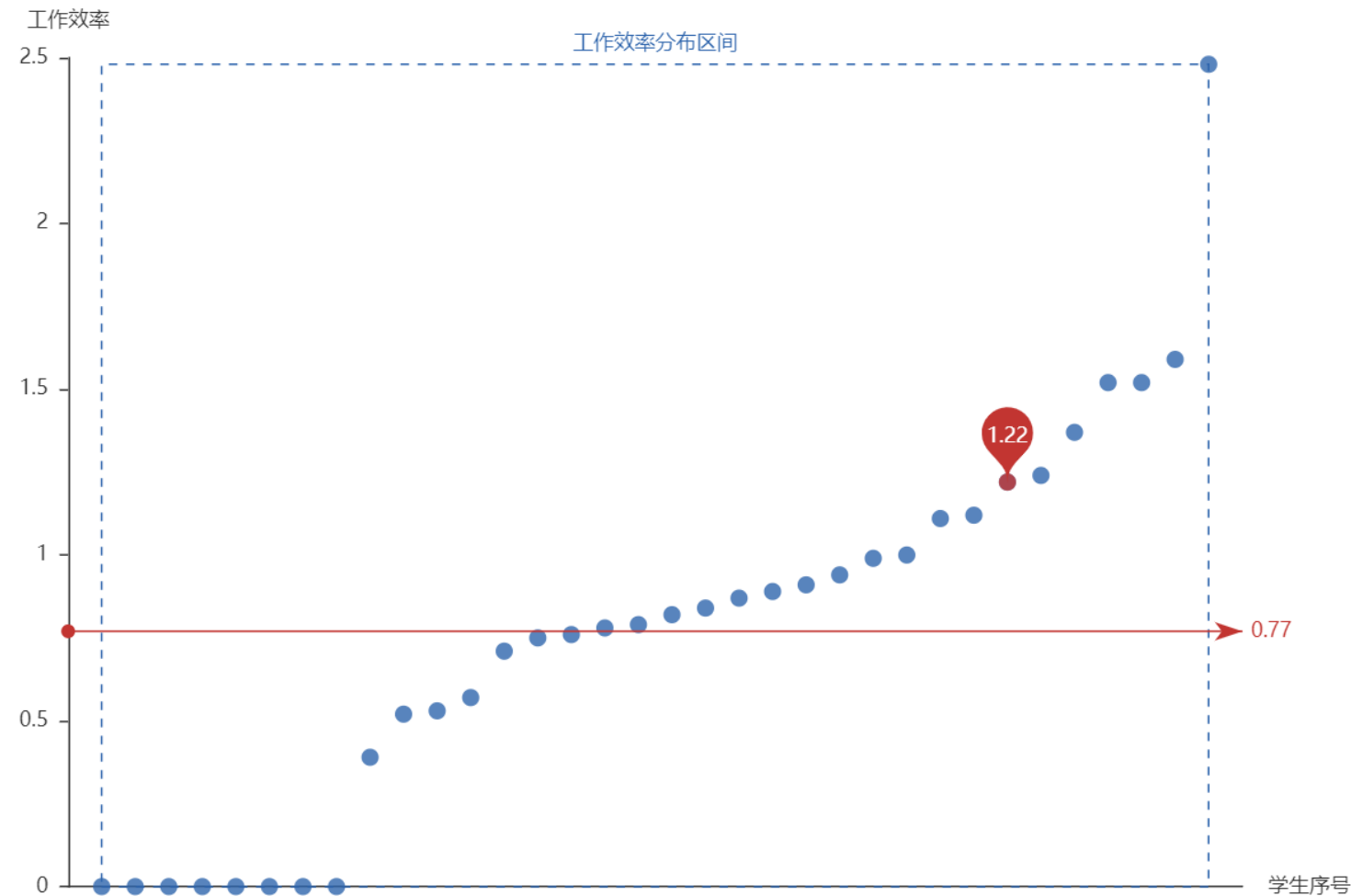


时间排序 | 成绩排序 | 学号排序 47 个检索结果 (47 学生)

序号	姓名	分班	学号	提交状态	提交耗时	更新时间	完成情况	最终成绩	操作
1		未分班	201606043018	延时提交	35天15小时3分	2019-05-14 23:58	2/2	80.0	调分 查看
2		未分班	201606021013	按时提交	10天21小时31分	2019-05-06 11:48	2/2	100.0	调分 查看
3	张 民	未分班	020021	延时提交	12天14小时17分	2019-04-21 23:12	2/2	80.0	调分 查看
4	谭 铭	未分班	201606020017	按时提交	12天12小时25分	2019-04-21 21:20	2/2	100.0	调分 查看
5	王 超	未分班	201606020005	按时提交	12天11小时16分	2019-04-21 20:11	2/2	100.0	调分 查看
6	廖 匀	未分班	020014	按时提交	11天7小时6分	2019-04-20 16:01	2/2	100.0	调分 查看
7	余 琪	未分班	201606020111	按时提交	11天0小时57分	2019-04-20 09:52	2/2	100.0	调分 查看
8	沈	未分班	201606020003	按时提交	10天2小时30分	2019-04-20 07:57	2/2	100.0	调分 查看

利用在线实训平台在开展课程实验

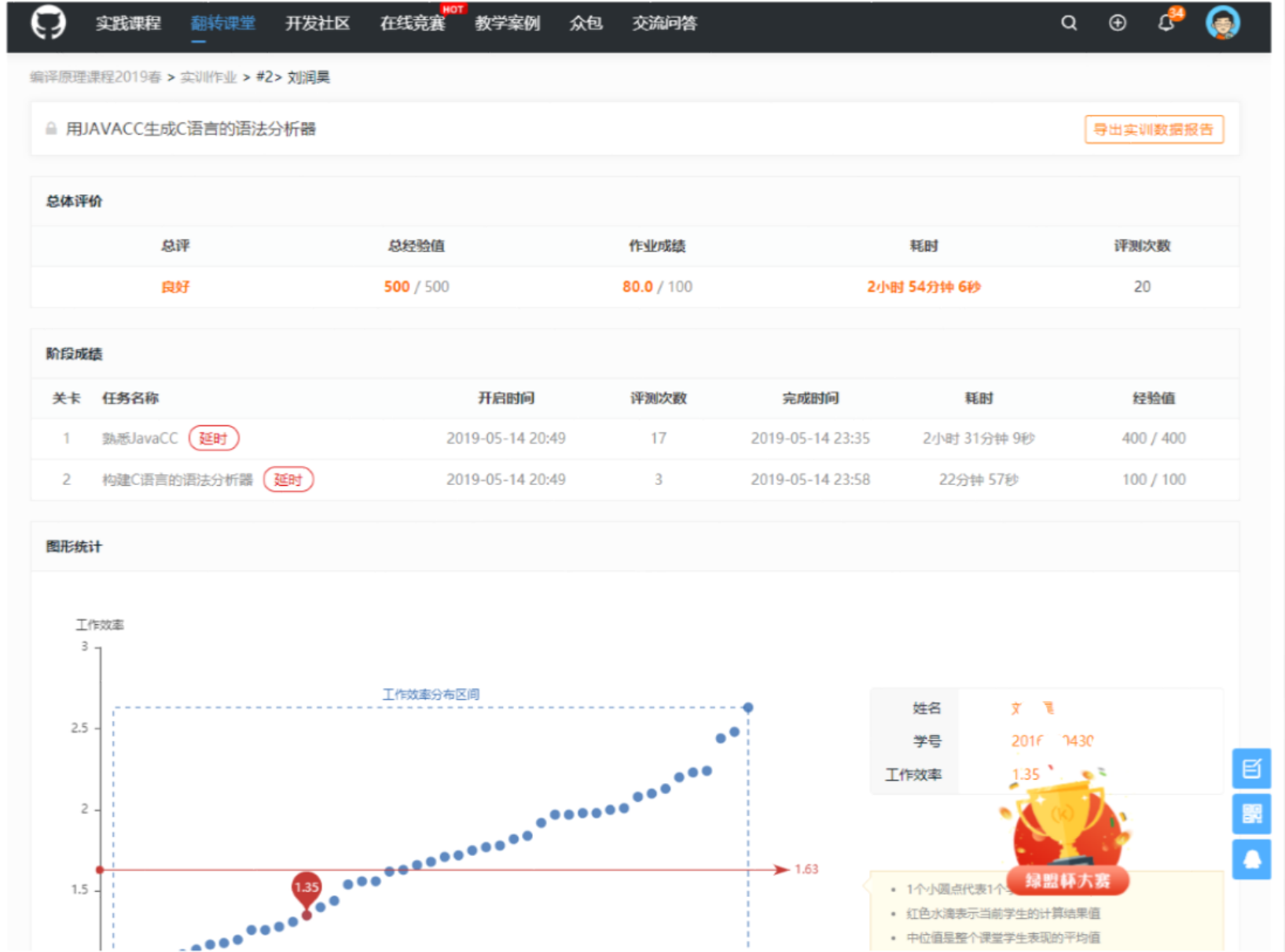
• 实验过程跟踪



评测次数	
第1次	共有5组测试集，其中有4组测试结果不匹配。
第2次	共有5组测试集，其中有4组测试结果不匹配。
第3次	共有5组测试集，其中有3组测试结果不匹配。
第4次	共有5组测试集，其中有3组测试结果不匹配。
第5次	共有5组测试集，其中有3组测试结果不匹配。
第6次	共有5组测试集，其中有3组测试结果不匹配。
第7次	共有5组测试集，其中有2组测试结果不匹配。
第8次	共有5组测试集，其中有2组测试结果不匹配。
第9次	共有5组测试集，其中有1组测试结果不匹配。
第10次	共有5组测试集，其中有5组测试结果不匹配。
第11次	共有5组测试集，其中有3组测试结果不匹配。
最后一次	compile successfully

利用在线实训平台在开展课程实验

- 自动生成实训报告(PDF)



编译原理课程2019春 > 实训作业 > #2> 刘润昊

用JAVACC生成C语言的语法分析器 导出实训数据报告

总体评价

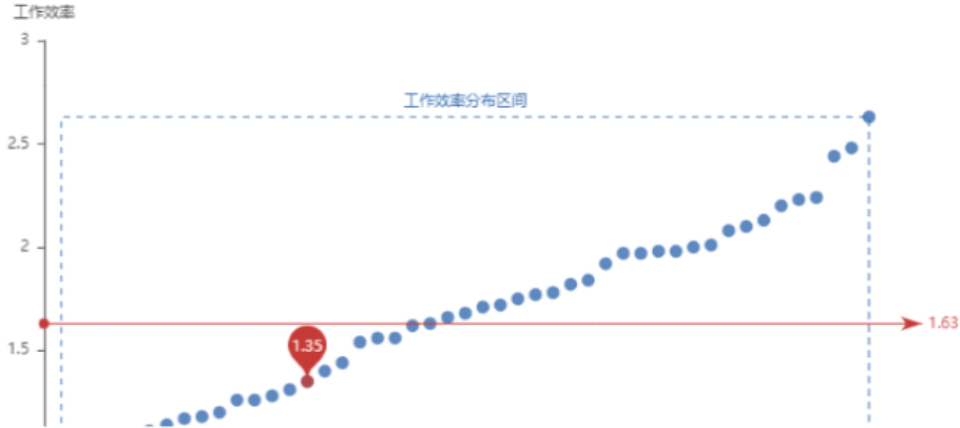
总评	总经验值	作业成绩	耗时	评测次数
良好	500 / 500	80.0 / 100	2小时 54分钟 6秒	20

阶段成绩

关卡	任务名称	开启时间	评测次数	完成时间	耗时	经验值
1	熟悉JavaCC (延时)	2019-05-14 20:49	17	2019-05-14 23:35	2小时 31分钟 9秒	400 / 400
2	构建C语言的语法分析器 (延时)	2019-05-14 20:49	3	2019-05-14 23:58	22分钟 57秒	100 / 100

图形统计

工作效率



姓名: 文 飞
学号: 2016 7430
工作效率: 1.35

绿盟杯大赛

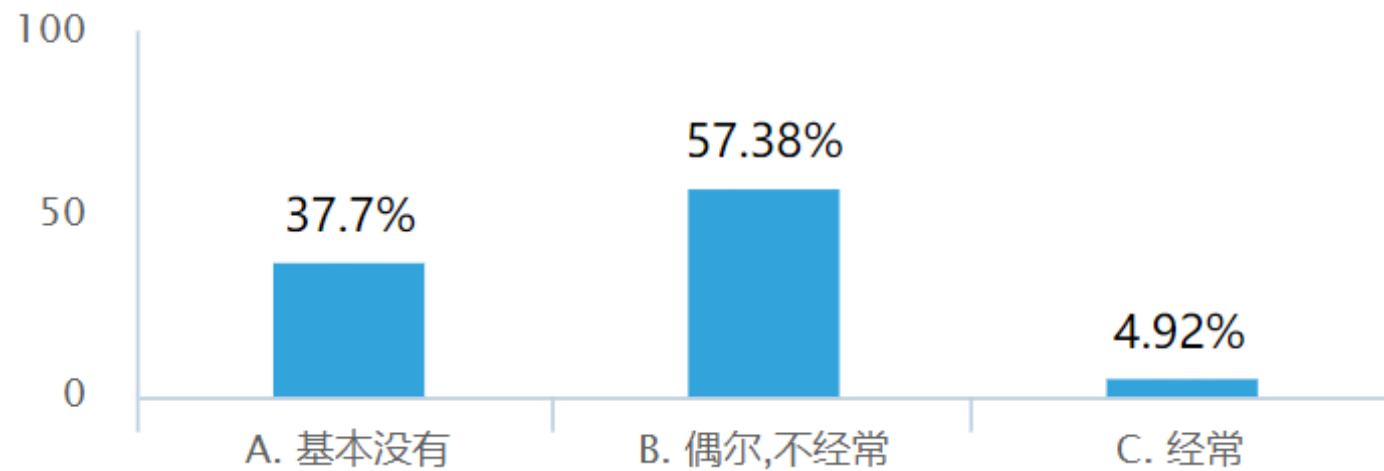
- 1个小圆点代表1个学生
- 红色水滴表示当前学生的计算结果值
- 中位值是整个课堂学生表现的平均值

课程调查

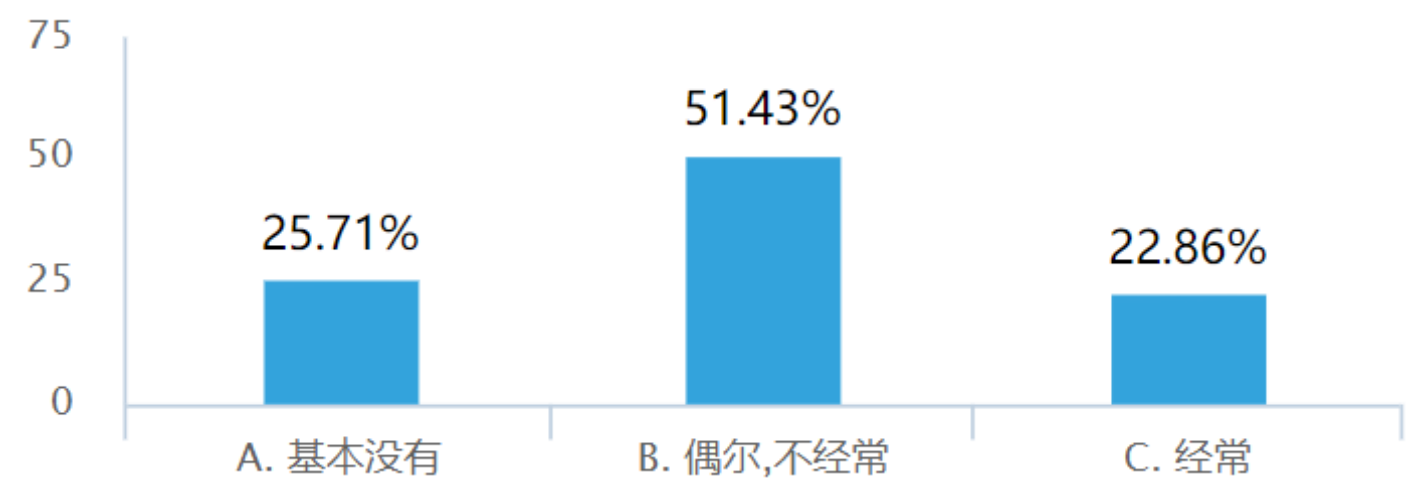
- 关于MOOC学习的调查
 - 你观看MOOC视频使用了哪些途径
 - 你在本课程学习过程中进行课前预习吗?
 - 你在本课程学习过程中进行课后复习吗?
 - 你觉得在本课程学习过程中课外自学效果如何?
- 关于在线实训项目的调查
 - 你觉得Educoder上的测试用例驱动开发的模式如何?
 - 你觉得Educoder上的实验项目设置难度梯度、逐级攻关, 效果如何?
 - 你觉得Educoder上开展实验项目最大的优势是什么?
 - 在单元实验项目中, 你是如何利用Educoder平台的?
 - 在综合实验项目中, 你是如何利用Educoder平台的?
 - 你觉得在Educoder开展编译课程实验, 你的实验学习的整体效果如何?
 - 你觉得在Educoder平台上开展实验作业是否对实验教学有帮助?

课程调查

你在课程学习过程中进行课前预习吗?(前)

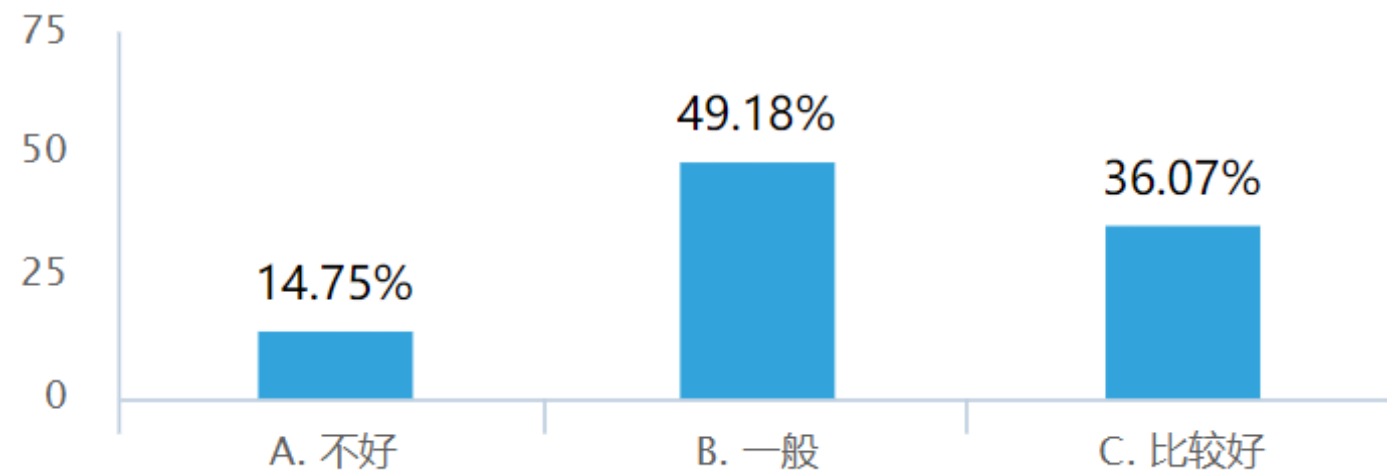


你在本课程学习过程中进行课前预习吗?(后)

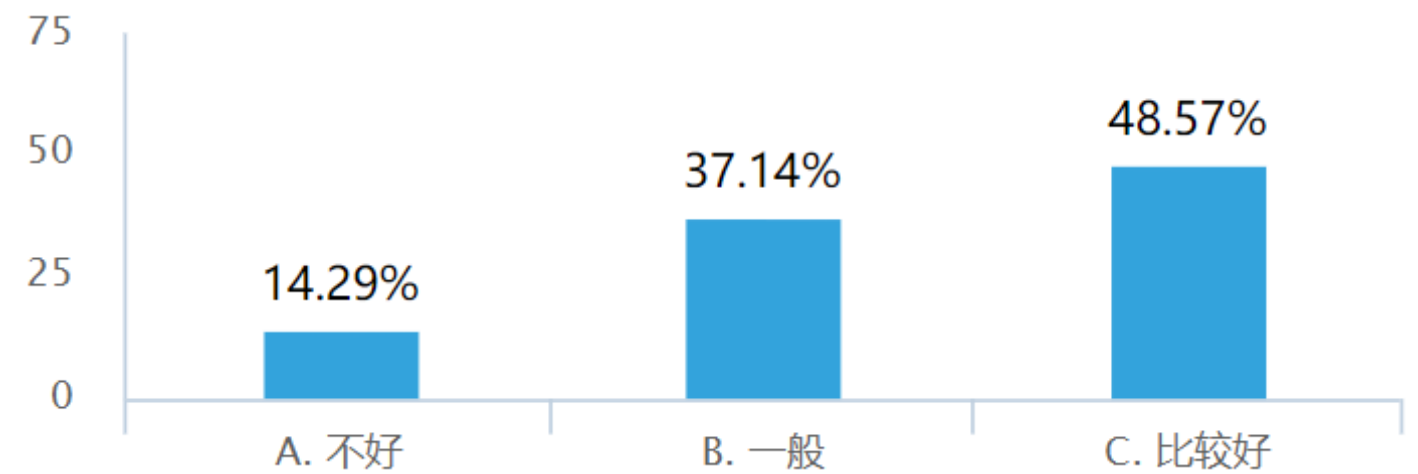


课程调查

你觉得在课程学习过程中课外自学效果如何?(前)

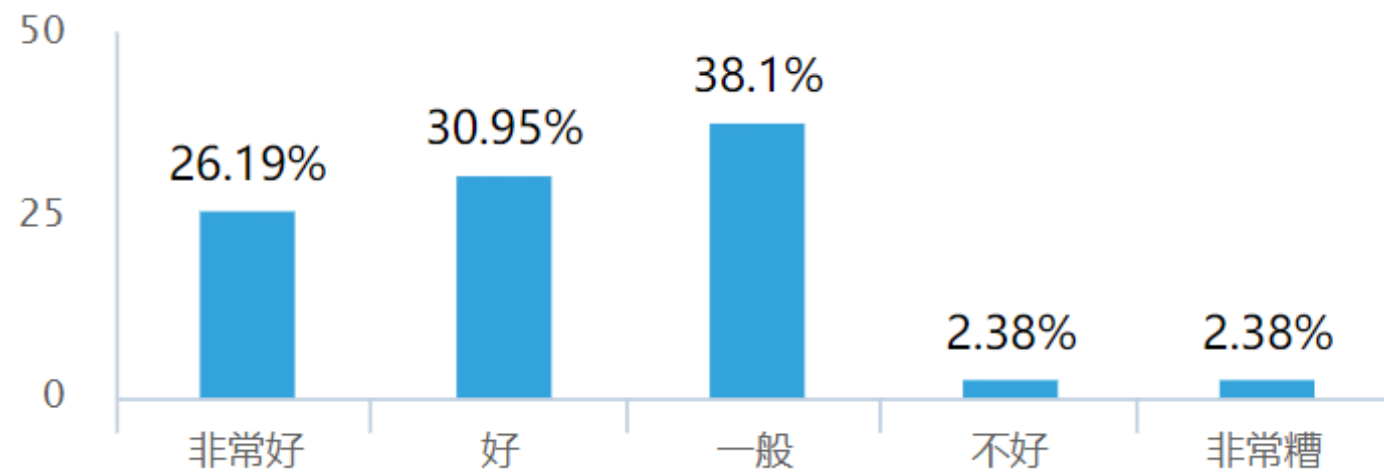


你觉得在本课程学习过程中课外自学效果如何?(后)

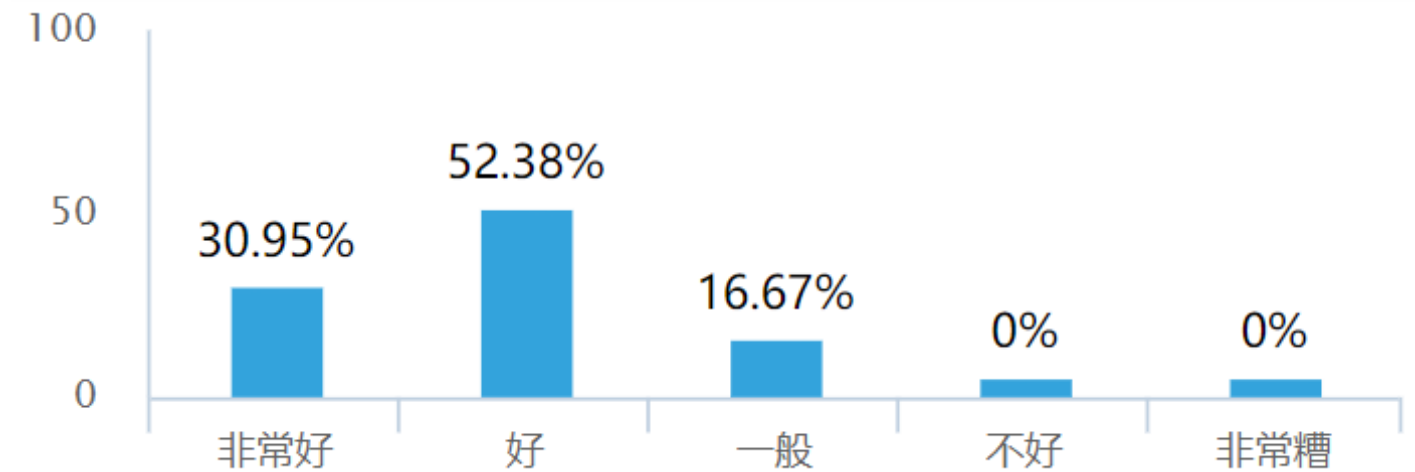


课程调查

你觉得Educoder上的测试用例驱动开发的模式如何？



你觉得Educoder上的实验项目设置难度梯度、逐级攻关，效果如何？



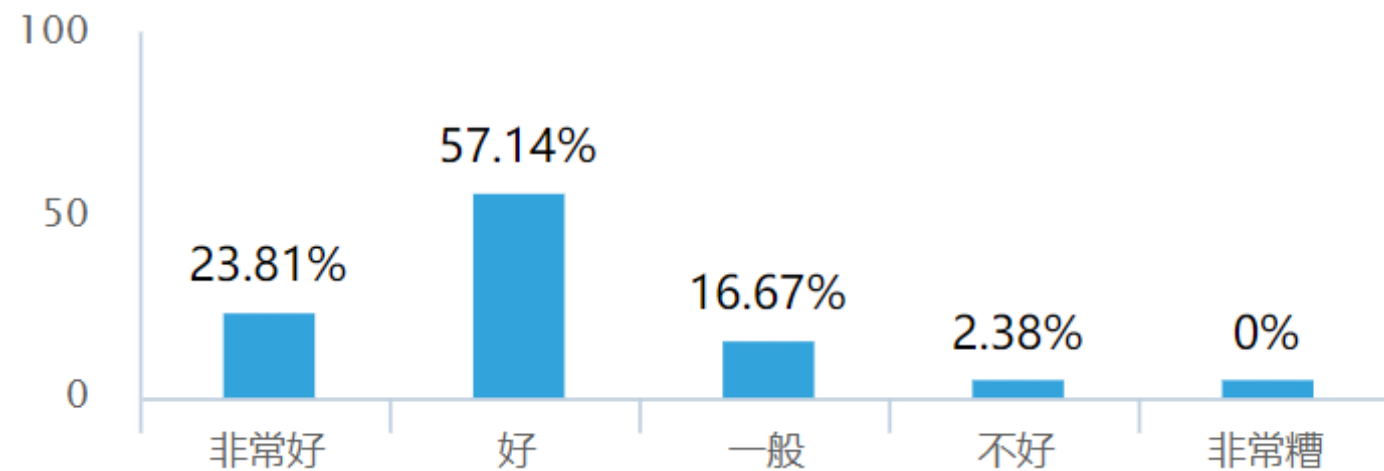
课程调查

你觉得Educoder上开展实验项目最大的优势是什么？

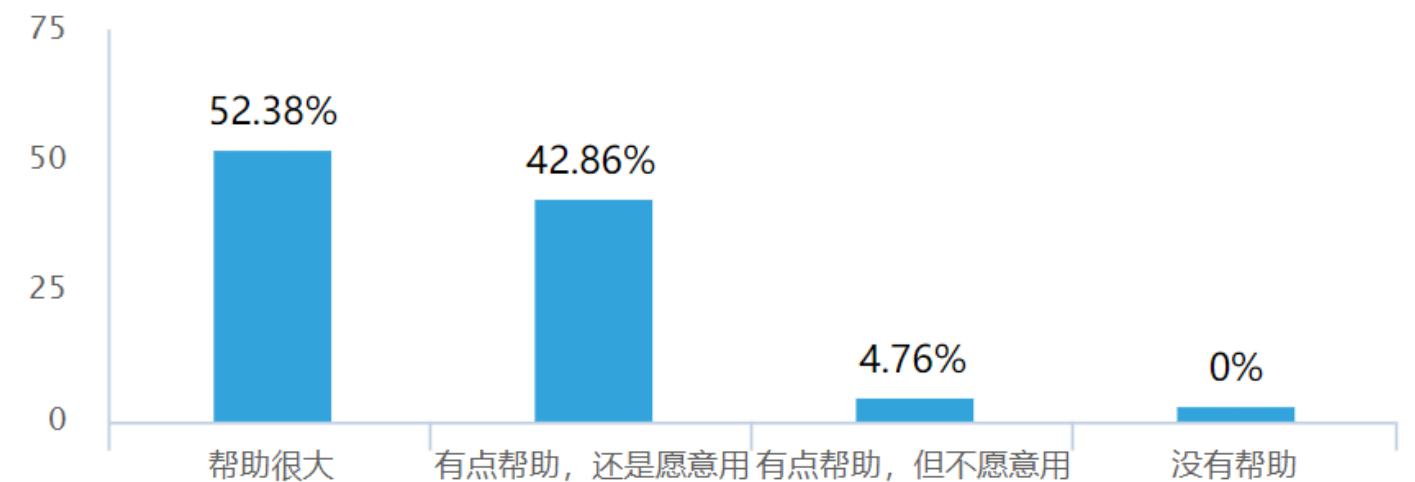
选项	小计	比例
从易到难有梯度	27	64.29%
便于看到自己的进展	26	61.9%
测试用例使得任务更明确	28	66.67%
平台模式利于讨论交流	6	14.29%
与其他课程建立联系，形成体系	6	14.29%

课程调查

你觉得在Educoder开展编译课程实验，你的实验学习的整体效果如何？



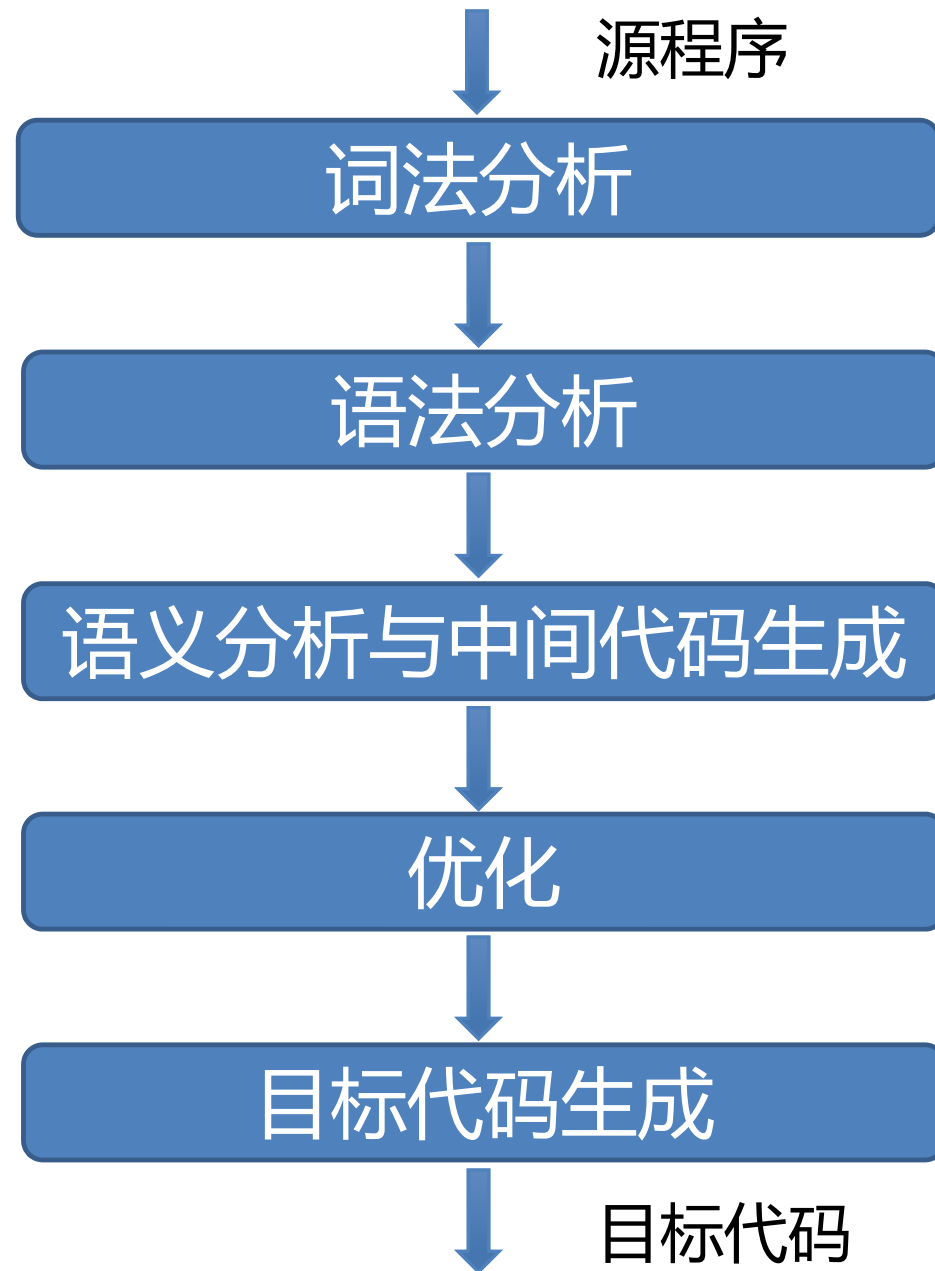
你觉得在Educoder平台上开展实验作业是否对实验教学有帮助？



目录

- 编译原理课程的定位
- 教学实践情况
- 课程下一步考虑
- 总结

编译课程



编译原理(本)

学时: 36

学分: 2

学时: 36

学分: 2

高级编译技术(研)

编译原理
学时: 80

课程下一步考虑

- 围绕复杂工程问题求解能力梳理教学内容
 - 保持课程的理论特色
 - 引入更多的编译技术进展：新的编程抽象、体系结构
- 以课程实验项目为切入点
 - 引入新的平台开展实践：LLVM、CLANG、...
 - 领域特定语言(DSL)设计实现

课程下一步考虑

- 高级编译技术(黄春研究员主讲)中的典型实验项目

Project : 语言实现

- 实现某领域的提出的语言或者语言扩展
 - ⊕ 基于flex或者yacc实现
 - ⊕ 或者手工实现
- Hints
 - ⊕ 采用源-源方式实现
 - 转换为C语言+库
 - ⊕ 直接转换为GCC/LLVM中间表示

课程下一步考虑

- 高级编译技术(黄春研究员主讲)中的典型实验项目

Project

Project : 语言实现

- 实现某领域的
 - ⊕ 基于flex或者
 - ⊕ 或者手工实现

■ Hints

- ⊕ 采用源-源方
 - 转换为C语
- ⊕ 直接转换为C

- 为提高程序可信性，通常使用C的安全子集，以避免一些“危险”语法。根据GJB5369，实现若干C安全子集规则，例如

- ⊕ 禁止循环中出现break;
- ⊕ 有符号位域长度<2
- ⊕ 静态函数未在所在的上下文中被调用
- ⊕ 使用++，或者—
- ⊕ 使用+=， -=， *=

■ Hints

- ⊕ 利用LLVM Clang 静态分析框架开发



课程下一步考虑

- 高级编译技术(黄春研究员主讲)中的典型实验项目

Project

- 实现某领域的
 - ⊕ 基于flex或者
 - ⊕ 或者手工实现

Hints

- ⊕ 采用源-源方
 - 转换为C语
- ⊕ 直接转换为C

Project

- 为提高程序可性
 - 以避免一些“介
 - 现若干C安全子
 - ⊕ 禁止循环中出
 - ⊕ 有符号位域长
 - ⊕ 静态函数未在
 - ⊕ 使用++, 或者
 - ⊕ 使用+=, -=, *

Hints

- ⊕ 利用LLVM Cl

Project: 优化算法改进

- 改进现有优化算法

- ⊕ 高级循环变换
- ⊕ 数据预取
- ⊕ Peephole优化
- ⊕ 标量优化

Hints

- ⊕ 阅读论文, 尤其是CGO相关论文, 在GCC/LLVM中实现
- ⊕ 阅读体系结构文档, 根据体系结构文档优化机器描述和其他参数配置

课程下一步考虑

- 高级编译技术(黄春研究员主讲)中的典型实验项目

Project

- 实现某领域的
 - ⊕ 基于flex或者
 - ⊕ 或者手工实现

Hints

- ⊕ 采用源-源方
 - 转换为C语
- ⊕ 直接转换为C

Project

- 为提高程序可性以避免一些“介”
实现若干C安全子
 - ⊕ 禁止循环中出
 - ⊕ 有符号位域长
 - ⊕ 静态函数未在
 - ⊕ 使用++, 或者
 - ⊕ 使用+=, -=, *

Hints

- ⊕ 利用LLVM Cl

Project

- 改进现有优
 - ⊕ 高级循环变
 - ⊕ 数据预取
 - ⊕ Peephole优
 - ⊕ 标量优化

Hints

- ⊕ 阅读论文, 中实现
- ⊕ 阅读体系结 述和其他参

Project : 并行编译

■ CUDA实现探讨

- ⊕ LLVM中包含了CUDA初步实现NVVM
 - 生成ptx, 运行时动态翻译

■ 评估LLVM CUDA实现

- ⊕ 完备性
- ⊕ 性能, 和NVCC对比

目录

- 编译原理课程的定位
- 教学实践情况
- 课程下一步考虑
- **总结**

总结

- 编译课程定位：通过抽象、理论指导求解复杂工程问题
- 编译课程教学
 - 利用经典理论成果吸引学生
 - 通过应用案例拓展学生的视野
- 下一步工作
 - 利用混合学习改进教学过程，提高学习效率
 - 围绕复杂工程问题求解能力梳理教学内容
 - 引入新的平台开展实践：LLVM、CLANG、...

感谢各位专家老师指正!