



中国科学技术大学
University of Science and Technology of China



2021全国大学生计算机系统能力大赛编译系统设计赛（华为毕昇杯）技术培训

以系统和创新能力培养 为导向的编译实践

中国科大计算机学院

李 诚

2021-04-17

- 中科大编译教学情况汇报
- 2020年参赛指导回顾
- 对同学们的建议



**张昱
老师**

教指委计算机系统专家委员会委员；2019年度高校计算机专业优秀教师奖励计划；2018年获安徽省教学成果二等奖；主讲编译原理H班课程



**郑启龙
老师**

参与编写《并行算法实践》，译著《编译器构造 - C语言描述》；国家级教学成果二等奖(09)；省教学成果特等奖(08)；主讲编译普通班课程



**李诚
老师**

国际计算机学会FCA成员，入选微软铸星计划。中科大青教赛二等奖(2020)；发表OSDI、FAST等论文；主讲编译普通班课程



**李卫海
老师**

信息安全专业副教授。研究方向包括多媒体内容安全、遥感信息处理等。主持国自然、国家863等项目。主讲网安编译课程。



**徐伟
老师**

实验教学中心高级实验师；指导学生获2020CCF大学生计算机系统与程序设计竞赛全国铜奖两项；参与编译课程实践体系建设

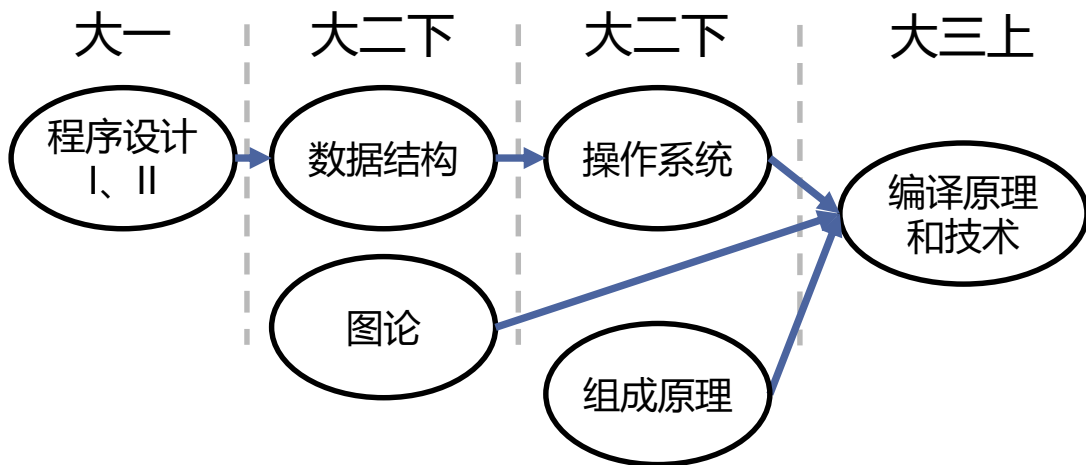
□ 中科大编译原理课程为安徽省省级精品资源共享课程；团队承担了6项教改项目，国家级1项，省部级3项；本团队教师获得了各类奖励15项，其中国家级6项，省部级2项。



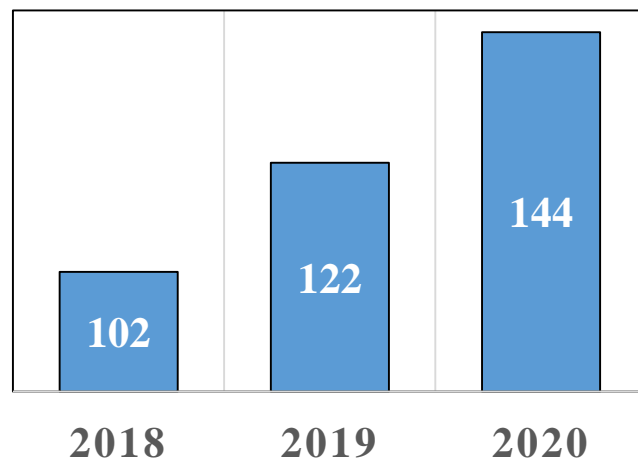
学情分析及教学目标



中国科学技术大学
University of Science and Technology of China



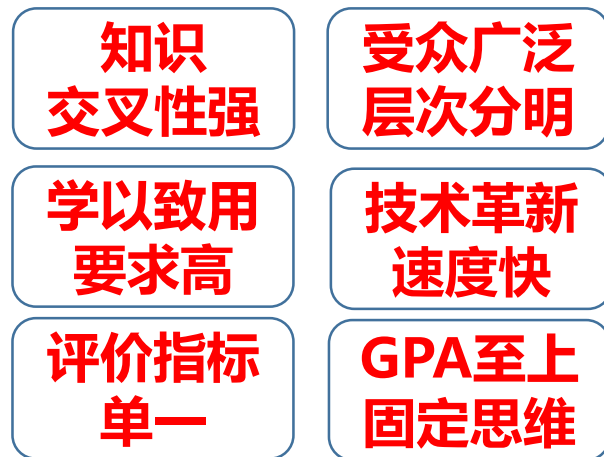
专业核心课及其诸多先导课程



选课人数逐年增多



课程教学目标艰巨



教学过程中面临很多挑战₄



□目标：以**系统和创新能力**的培养为导向，以“**金课**”标准为指导思想，根据计算机学科发展的趋势和新时代学生的成长需求构建**现代编译实验体系**，体现**先进性、挑战度**，增强学生**获得感**

源语言特性

以Cminus语言为基础，增加对浮点数、数组等复杂特性的支持

词法语法 分析器

使用Flex+Bison工具构造编译器前端，同时鼓励使用ANTLR，重点考察对Cminus语言正则表达式和语法产生式的抽象能力

中间代码 生成器

使用简化版LLVM中间表达子模块和基于访问者模式的编程框架(**自研**)，编写生成器，自动化生成 Cminus程序的中间代码

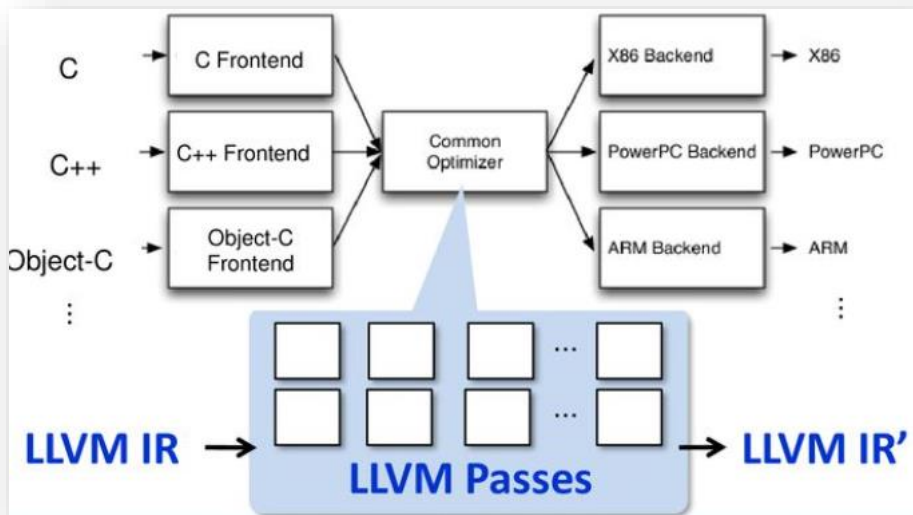
中间代码 优化器

使用封装好的基本块、流图、前驱后继结点、调用者使用者链表等常用数据结构及其接口(**自研**)，实现经典的代码优化算法

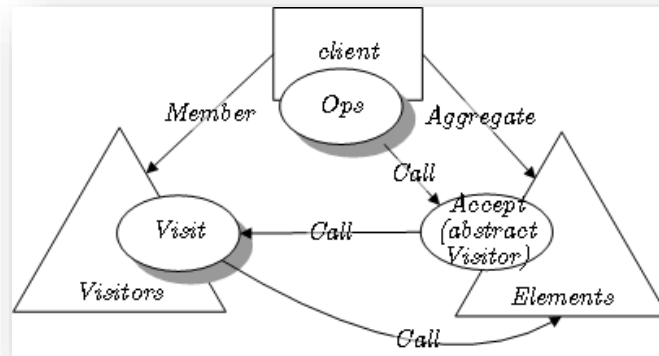
目标代码 生成器

提供RISC-V、MIPS等多指令集后端，鼓励对寄存器分配算法、指令选择算法等进行探索验证，学会用模拟器评测目标代码

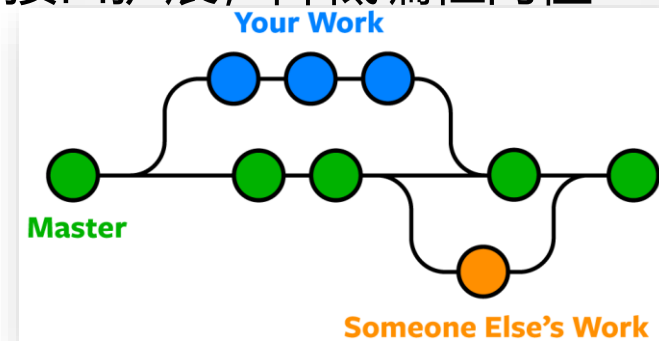
紧跟时代前沿，融合先进技术，架起课程教育与工业界技术革新之间的桥梁



在全国范围内较早地将**LLVM——这一工业界广泛使用的、产品级现代编译工具链**引入到课程实验中，建立贴近业界需求、适合多种教学目标的循序渐进的课程实践体系



使用适合编译器编写的**访问者设计模式**，将数据结构与操作分离开来，便于接口扩展，降低编程门槛

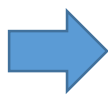


采用**git进行版本管理**，便于学生的敏捷开发、项目管理和协同工作



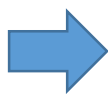
□合理增加实验难度、拓展实验深度，体现课程的高阶性、创新性和挑战度

编程与
工程实践能力



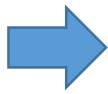
- 5次大实验，要求学生独立完成至少**1000**行的C++代码
- 要求学生为每次实验撰写**工程设计和测试文档**，完整记录全过程

自主学习能力



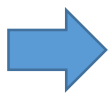
在16周内，要求学生从零开始，学会使用**C++ STL、LLVM、Flex、Bison、Clang、Git、GDB、VM、Docker**等**数十种新编程工具和系统软件**

领导与
团队协作能力



- 安排2次分组实验，**对组长和组员进行分别考核**，衡量贡献比
- 要求学生熟练使用Git进行**合作开发**

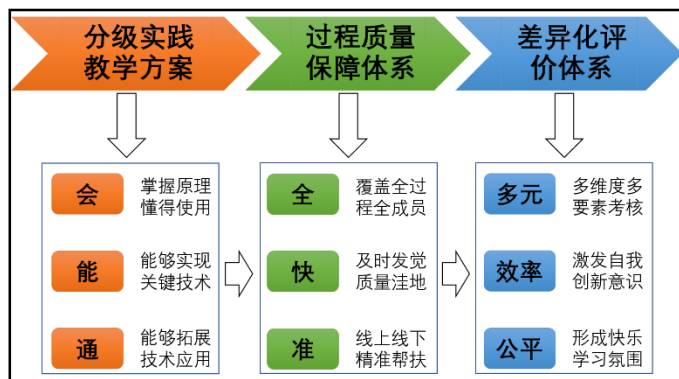
探索与创新能力



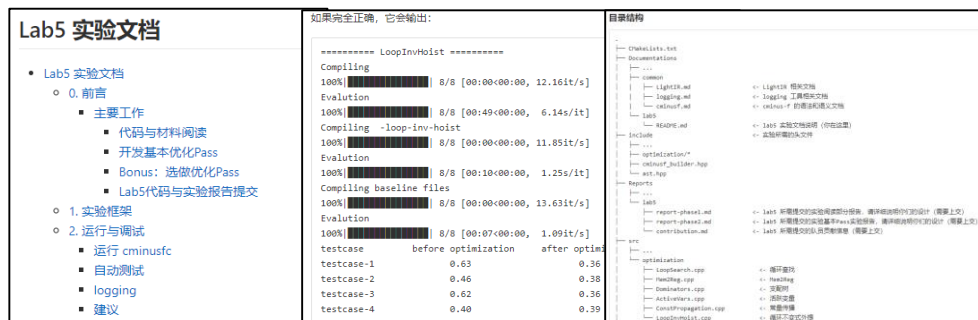
通过分级实验设计和加分机制，鼓励学有余力的学生选做更有挑战的项目，比如：**函数内联，尾递归消除...**



结合学生差异化需求，降低实践入门难度，设计递进式实践教学方案及过程质量管理和多元评价体系



坚持分级实验+过程管理+多元评价结合的指导思想，让每一位学生都**学有所思、学有所乐、学有所得**



- 自主编写了45000余字的实验文档，自主设计了简化版的Light-LLVM-IR框架，明确实验目的和考核标准，降低了工程实践的门槛
- 将挑战度高的实验(如代码优化)拆解为若干层次分明、衔接有序的小实验，及时反馈结果
- 组建8人制助教团队，形成了助教轮值制度，将学生问题的反馈时间压缩到12小时以内，获得学生广泛好评。



□结合学生差异化需求，降低实践入门难度，设计递进式实践教学方案及过程质量管理和多元评价体系



基于GitLab建设了活跃的学习社区

建设成效

- 2020年**提交率达到94.4%**，在选课人数增加了18.0%的前提下，相较2019年**提高了6.3%**。
- 增强**学生的主体意识**，2020年，144名同学提交258个内容丰富的gitlab issues，其中61名同学获得额外加分。
- 以lab5为例，**57组中12组参加了高阶(非必须)实验**，完成质量有区分度，达到拔高目的。
- 2019年，1组同学**发现一个隐藏在LLVM中的bug**，获得官方人员的认可，该bug随后被修复。



科教融合

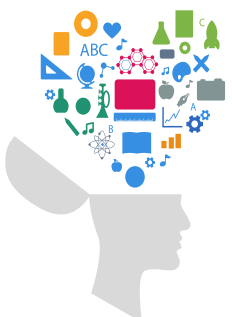


• 引研入教


• 以教促研

• 激发创新


系统全面

- 
- 操作系统
 - 体系结构
 - 编程语言
 - 算法分析

重视实践

- 
- 使用新技术
 - 实验迭代
 - 敏捷开发
 - 项目管理

多维考核

- 
- 分级实验
 - 学习社区
 - 合作协同
 - 表达能力

□ 中科大编译教学情况汇报

□ 2020年参赛指导回顾

□ 对同学们的建议

□指导教师

❖李诚、李卫海

□队伍1：燃烧我的编译器

❖陈清源、黄奕桐、章耀辉、曾明亮

□队伍2：Yet Another LLVM

❖丁伯尧 刘硕 黄业琦 彭昀

□学生来源：

❖计算机学院、网安学院、少年班学院等

□指导教师

❖李诚、李卫海

□队伍1：燃烧我的编译器(第一名、特等奖)

❖陈清源、黄奕桐、章耀辉、曾明亮

□队伍2：Yet Another LLVM(第八名、二等奖)

❖丁伯尧 刘硕 黄业琦 彭昀

□学生来源：

❖计算机学院、网安学院、少年班学院等

□指导教师

❖李诚、李卫海

行云流水、步步为营

□队伍1：燃烧我的编译器(第一名、特等奖)

❖陈清源、黄奕桐、章耀辉、曾明亮

□队伍2：Yet Another LLVM(第八名、二等奖)

❖丁伯尧 刘硕 黄业琦 彭昀

跌宕起伏、力挽狂澜

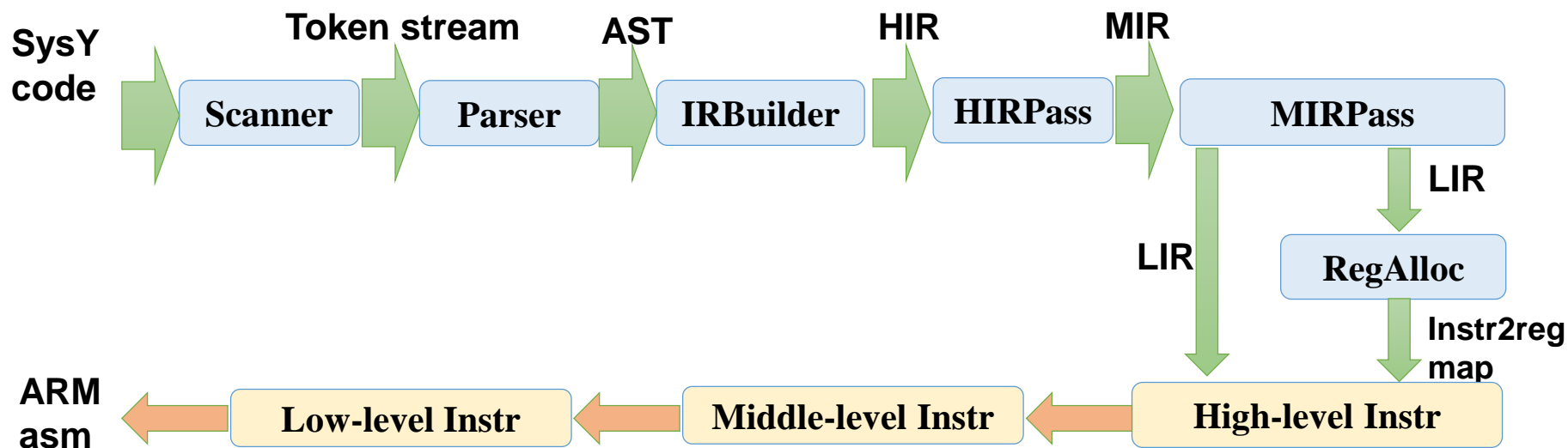
□学生来源：

❖计算机学院、网安学院、少年班学院等



三个层次的 IR

- HIR 保留代码 if 与 while 等结构, 方便结构级变换
- MIR 简洁优雅的语义, 承载大量通用优化 Pass
- LIR 贴近硬件架构, 便于指令融合、调度与选择



三个层次的虚拟指令

- HInstr 智能翻译成多条 MInstr, 不可使用临时寄存器
- MInstr 智能翻译成多条 LInstr, 可用临时寄存器 R12 R14
- LIInstr 智能翻译成多条 ARM Instr, 可用临时寄存器 R11



□在有限的时间里实现了30+优化pass

HIR

- LoopMerge
- AccumulatePattern
- HighIRsimplyCFG
- BBExps
- MergeCond
- ...

LIR

- InstructionSchedule
- InstructionFuse
 - SpliGEP
 - SplictRem
 - removeUnuseOP
 - convertMulDivToShift
- ...

MIR

- IRCheck
- LoopFinder
- Dominator
- ActiveVar
- ReachDefinition
- Mem2reg
- BBConstPropagation
- DeadCodeEliminate
- FunctionInline
- SimplifyCFG
- LoopInvariant
- Mutithreading
- Vectorizaiton
- PowerArray
- ...

Analysis

Transform



队伍1：中间代码的极致化简



中国科学技术大学
University of Science and Technology of China

ConstPropagation

```
; Module name: 'SysY code'
define i32@main()
<label>entry      ;preds:
    [ 4 x i32]* %19 = Alloca Init 0
    i32* %20 = GEP [ 4 x i32]* %19 i32 3
    Store i32 1 i32* %20
    Br <label> %8
<label>8          ;preds: %entry
    Br <label> %11
<label>11         ;preds: %8
    Br <label> %15
<label>12         ;preds:
    Br <label> %15
<label>15         ;preds: %11 %12
    i32 %21 = PHI i32 2 <label> %11 i32 3 <label> %12
    i32* %1 = GEP [ 4 x i32]* %19 i32 %21
    i32 %2 = Load i32* %1
    ret i32 %2
```

SimplyCFG

```
; Module name: 'SysY code'
define i32@main()
<label>entry      ;preds:
    [ 4 x i32]* %19 = Alloca Init 0
    i32* %20 = GEP [ 4 x i32]* %19 i32 3
    Store i32 1 i32* %20
    i32* %1 = GEP [ 4 x i32]* %19 i32 2
    i32 %2 = Load i32* %1
    ret i32 %2
```

LIR Passes

```
; Module name: 'SysY code'
define i32@main()
<label>entry      ;preds:
    [ 4 x i32]* %19 = Alloca Init 0
    Store i32 1 [ 4 x i32]* %19 i32 12
    i32 %2 = Load [ 4 x i32]* %19 i32 8
    ret i32 %2
```



- 经典的多线程实现：每个线程有独立的栈空间
- 独创的多线程实现：每个线程共享部分栈空间
 - ❖ 更高的性能
 - ❖ 更方便变换
 - ❖ 更易于实现



队伍1：独创多线程框架



中国科学技术大学
University of Science and Technology of China

```
int my_func(.....) {
```

```
    ..... some context on stack .....
```

```
    for (int i = 0; i < n; i++) {  
        for (int j = 0; j < n; j++) {  
            for (int k = 0; k < n; k++) {  
                A[i][j] = A[i][j] + B[i][k] * C[k][j];
```

```
            }  
        }  
    }
```

```
}
```

```
.....
```

```
}
```

Call

pthread_create()

```
int matrix_mul(int t, .....(context).....) {  
    for (int i = starti(t); i < endi(t); i++) {  
        for (int j = 0; j < n; j++) {  
            for (int k = 0; k < n; k++) {  
                A[i][j] = A[i][j] + B[i][k] * C[k][j];  
            }  
        }  
    }  
}
```

经典的多线程实现（包装成函数）

```
int t = MTSTART();  
for (int i = starti(t) ; i < endi(t) ; i++) {  
    for (int j = 0; j < n; j++) {  
        for (int k = 0; k < n; k++) {  
            A[i][j] = A[i][j] + B[i][k] * C[k][j];  
        }  
    }  
}  
MTEND(t);
```

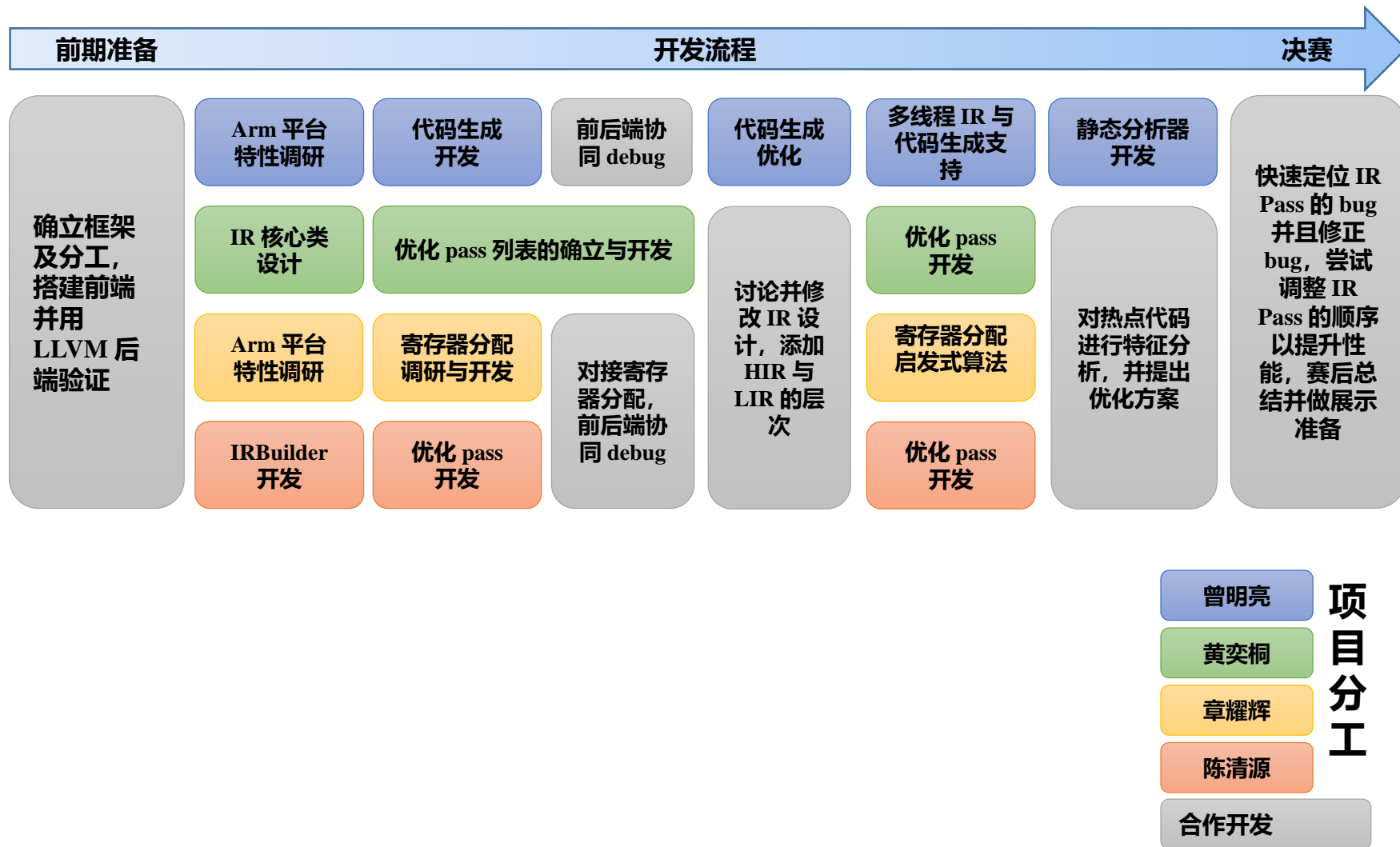
独创的多线程实现（就地完成）



队伍1：开发进度图



中国科学技术大学
University of Science and Technology of China

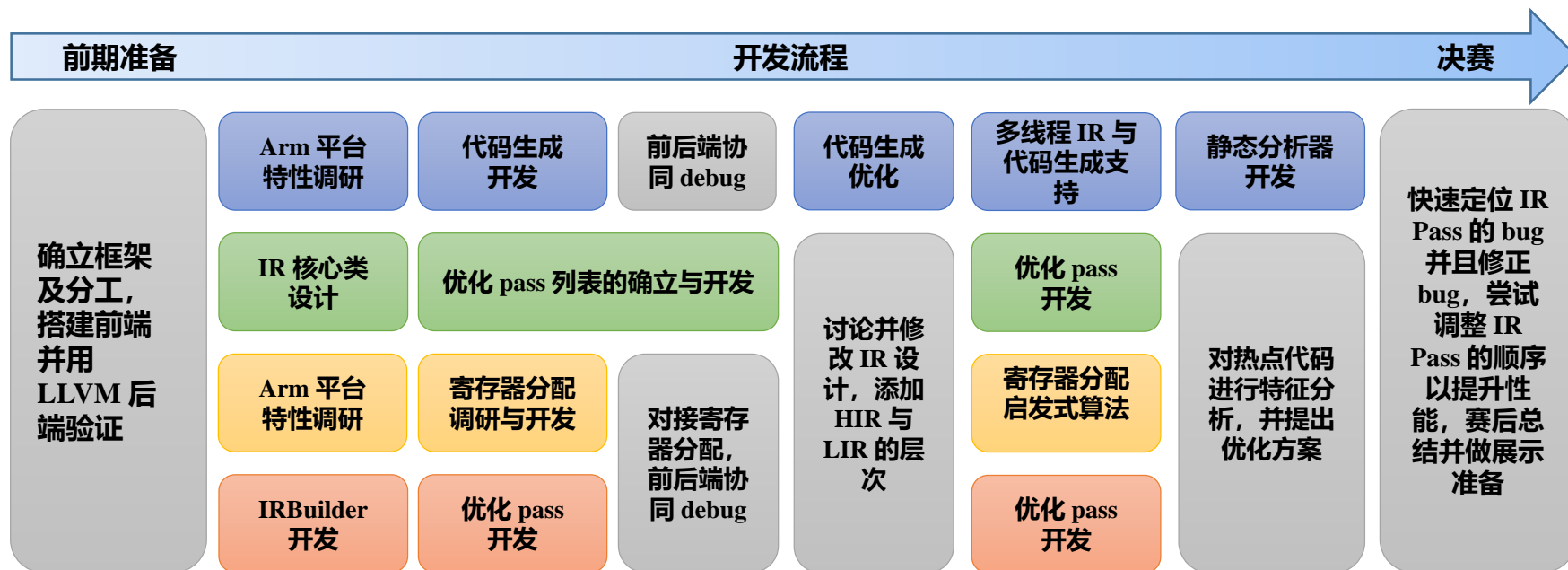




队伍1：开发进度图



中国科学技术大学
University of Science and Technology of China



时间节点:

- 7.22 成功通过了所有功能测试用例
- 7.23 经过一天的优化性能就超过了 gcc -O0
- 8.7 性能已经超过了 gcc -O2
- 8.21 在决赛当天达到了 gcc -O2 性能的四倍

注：所有结果仅限于大赛所给样例

曾明亮

黄奕桐

章耀辉

陈清源

合作开发

项目分工



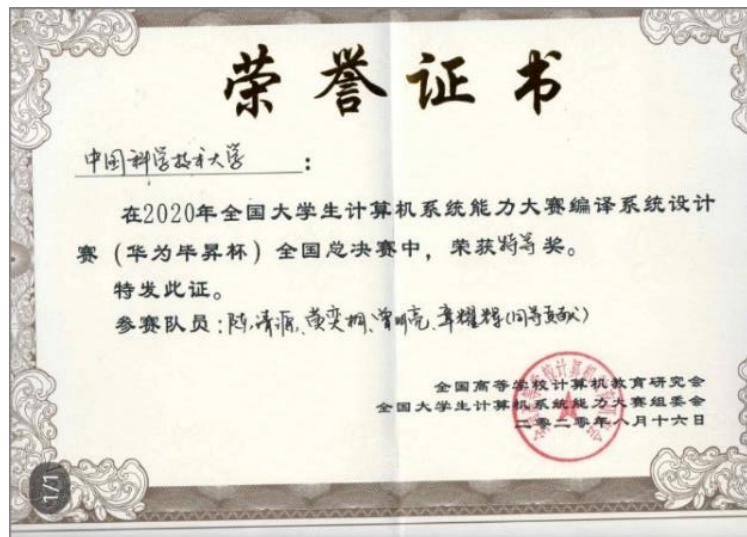
队伍1：专家点评



中国科学技术大学
University of Science and Technology of China

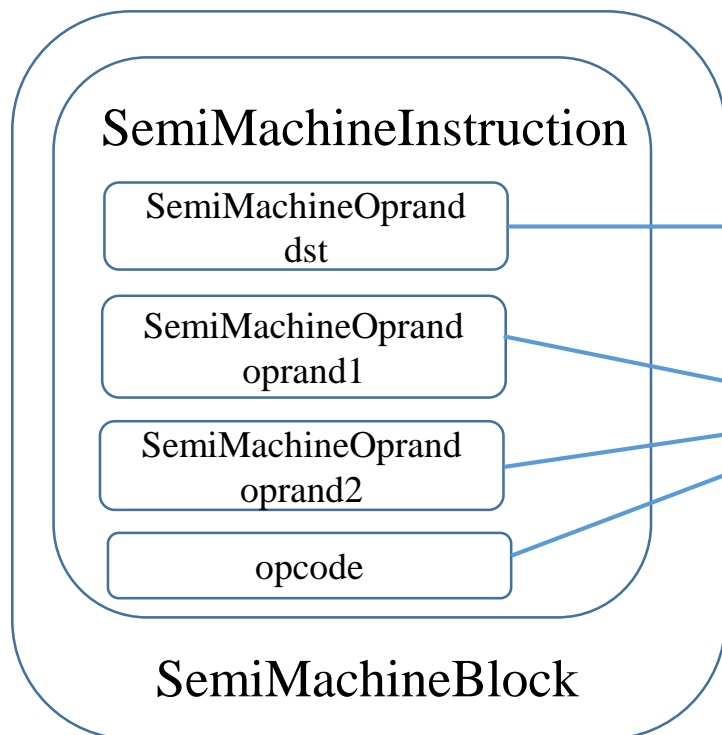


陈文光 清华大学计算机系教授，兼任青海大学计算机系主任。主要研究领域为操作系统、程序设计语言与并行计算。获国家科技进步二等奖一次，部级科技一等奖两次。现为中国计算机学会杰出会员和杰出讲者，副秘书长，青年科技论坛荣誉委员；ACM中国理事会共同主席。



编译大赛陈文光评委点评：

“自动向量化和自动多线程优化是编译优化中非常复杂的部分，**中科大队伍**开发的这些优化展现出了他们在编译基础知识上的掌握深度和积极主动的钻研态度，这也是**中科大**在编译和程序设计语言方面**高水平教学的直接体现**。”



SemiMachine IR设计的目的是为了
方便进行基于目标代码的优化

快速确认目标

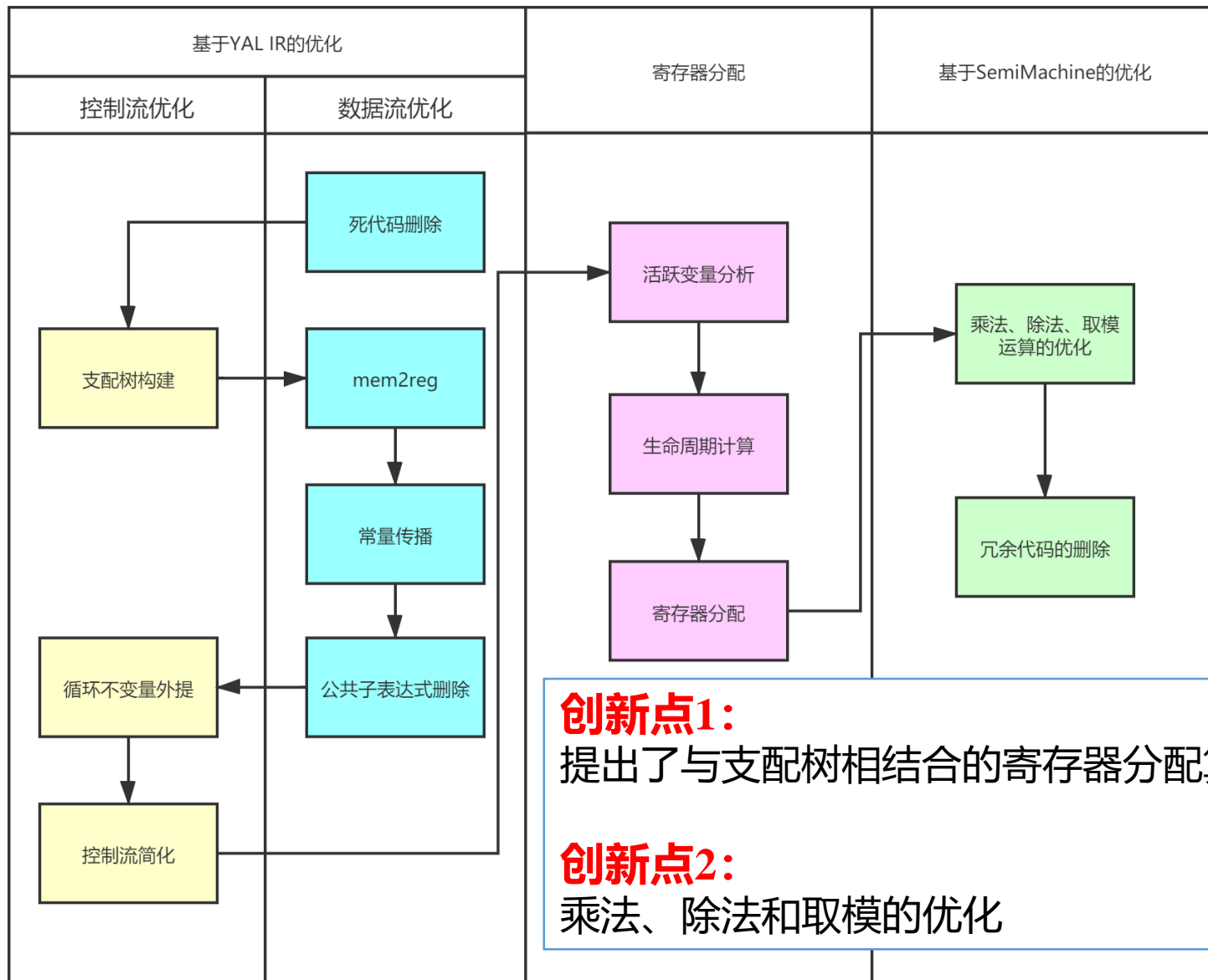
**快速确认操作数类型 (常数、
寄存器.....)**
便于做指令选择和乘除法优化



队伍2：优化pass



中国科学技术大学
University of Science and Technology of China





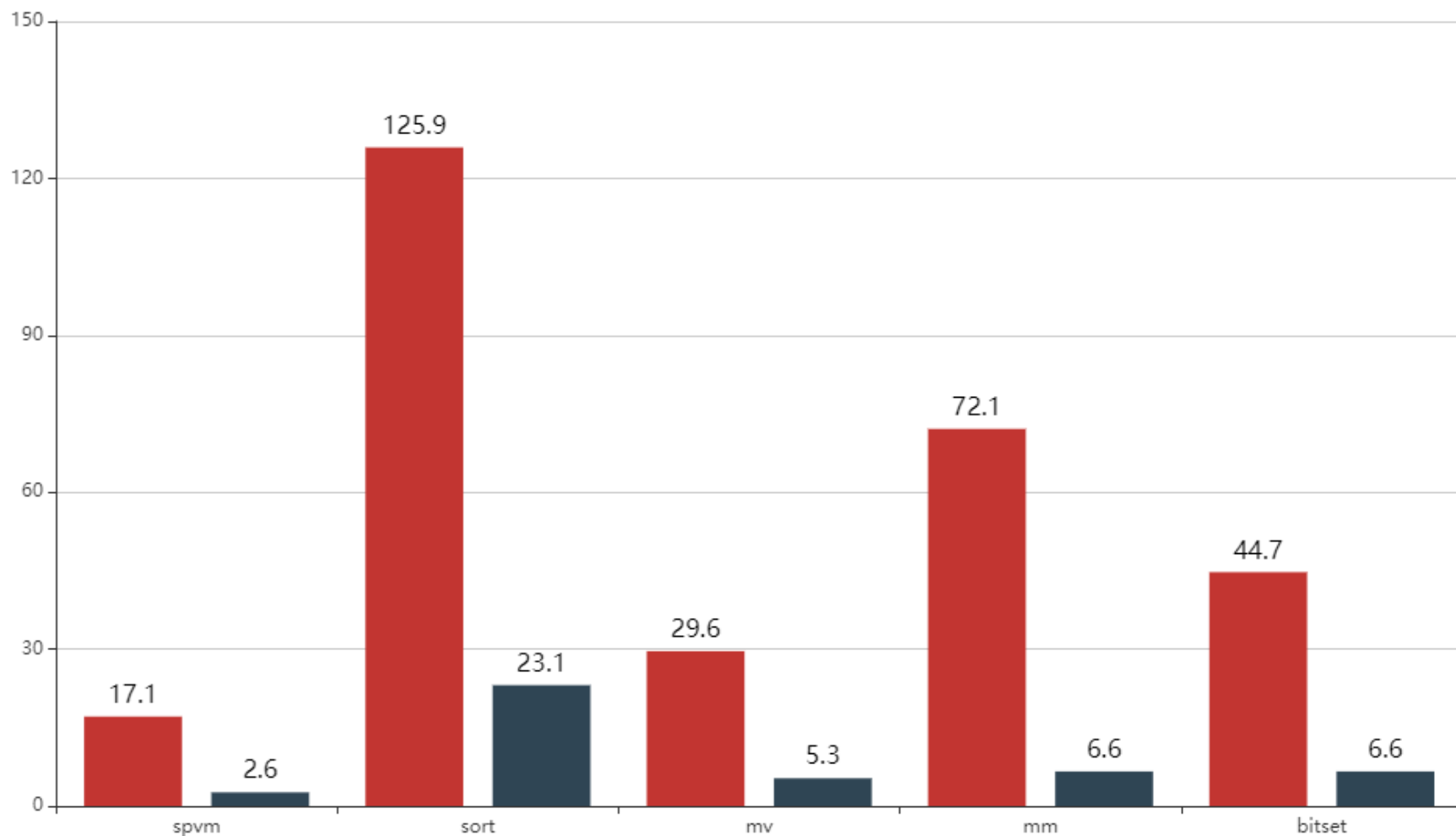
队伍2：优化性能效果



中国科学技术大学
University of Science and Technology of China

效果展示

■ 优化前 (秒) ■ 优化后 (秒)



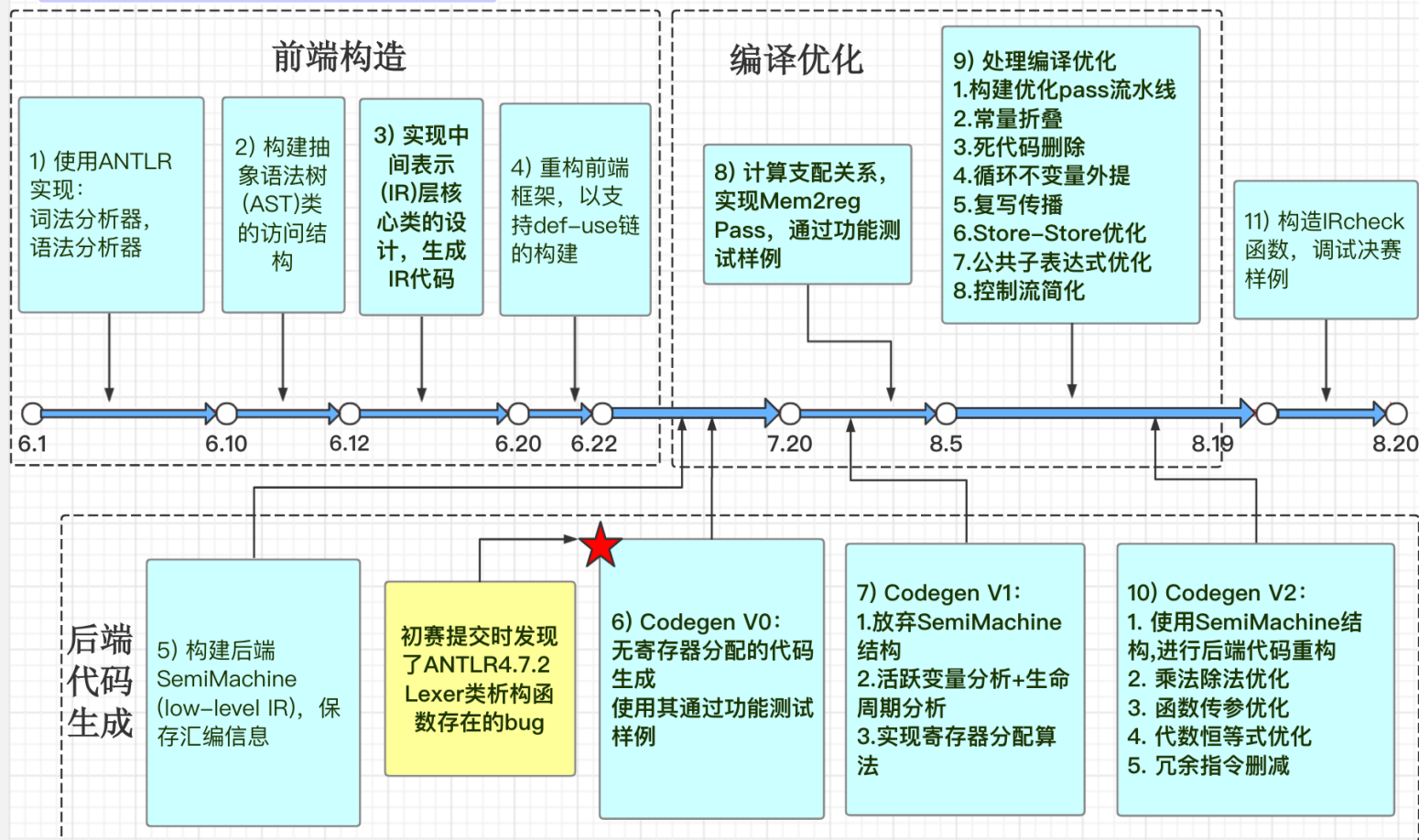


队伍2：开发进度图



中国科学技术大学
University of Science and Technology of China

YAL开发时间轴



- 中科大编译教学情况汇报
- 2020年参赛指导回顾
- 对同学们的建议**

□能力要求

- ❖队伍构成需要差异性，每个人都要有自己的特长，组员之间要有很强的互补性
- ❖领导能力与团队协作同等重要
- ❖时间要有保障，执行能力要强
- ❖抗压能力

□定期召开讨论会（每周一次，甚至两次）

- ❖每次会前准备条理清晰的ppt，重点讲设计思想、所遇到的问题
- ❖每次会上要积极参与讨论，表达观点
- ❖每次会后要及时总结讨论内容，给出todo list

□项目管理与推进

- ❖ 前两周的工作对于技术路线选定、重要时间节点的确
定、项目分工等很重要
- ❖ 项目管理能力很重要，组长需要明确阶段性任
务，要掌控整个项目的进展

□系统设计需要一个全局、模块化的思想，认真思考每一个模块的基础功能、扩展潜力等

□Keep it simple, stupid but working

- ❖ 尽早完成一个功能正确、可测试的版本
- ❖ 队伍1在7.22完成，但队伍2在初赛前两天才完成

□创新是考核的重点，也是性能突破的关键

❖改变课堂学习单向输入的思维定式

➤进行自我探索

❖突破知识学习的边界

➤科研论文？技术论坛？

❖对潜在解决方案优劣的甄别能力

➤流程分析？建模分析？

❖敢为天下先的勇气

➤队伍1甚至提出要开发硬件中GPU的资源



中国科学技术大学
University of Science and Technology of China



谢谢!

希望各位老师、同学享受比赛、取得优异成绩、并投身基础系统软件的伟大事业中!

中国科大计算机学院

李 诚

2021-04-17