

# 编程语言的设计原理 课程项目

胡振江、赵海燕、熊英飞

2021

# 课程项目要求

- 设计一个带类型系统的程序语言，解决实践中的问题，给出基本实现
- 要求：
  - 需要说明该语言解决实践中的什么问题
  - 需要论证该语言是否解决了该问题，解决的程度如何
  - 根据需要，给出类型系统的Safety的证明、类型检查算法和算法终止性证明
  - 根据需要，给出解决问题的形式化定义，并证明解决了该问题
  - 需要给出该语言的实现和示例程序。
    - 可以不给出语言的lexer和parser，示例程序可以直接用抽象语法树表达。通过使用OCaml或者Haskell等表达能力强的语言，抽象语法树可以和实际程序很类似。
    - 可以通过修改本课程给出的Ocaml代码实现。

# 课程项目要求

- 组队：2人为一个小组完成，特殊情况也可1-3人
- 考核时间点：
  - 4月12日：报组队名单给助教
  - 4月28日：选题报告。每组10分钟左右，介绍选题的意义和目标
  - 6月16日：口头报告。每小组20分钟左右，具体时间根据分组情况决定。
  - 6月20日：书面报告和代码提交。书面报告包括语言的设计目标、设计思想、形式定义、性质证明、代码使用方法等。

# 课程项目示例

- 设计一个语言，保证永远不会发生内存/资源泄露。
- 设计一个汇编语言的类型系统
- 设计一个没有停机问题的编程语言
- 设计一个嵌入复杂度表示的类型系统，保证编写的程序的复杂度不会高于类型标示的复杂度。
- 设计一个类型系统，使得敏感信息永远不会泄露。
- 设计一个类型系统，使得写出的并行程序没有竞争问题
- 设计一个类型系统，保证所有的浮点计算都满足一定精度要求（比如：保留若干有效数位）
- 解决自己研究领域的具体问题

# 往年项目举例

- 2017 陈逸凡、吴昊泽、吉如一。针对不可靠输入数据的编程语言。
  - 假设输入会发生随机采样错误，给定输出的可靠性，自动算出输入所需的采样次数。
- 2016 王然、王迪。设计了一个针对下标的Dependent Type系统，可以在相当丰富的程序类型上完成如下检查，能力超过一些近期论文里的系统
  - 数组越界检查
  - 函数作用域检查
  - 矩阵运算维度检查
- 2018 王伦。设计了一个基于差分隐私的编程语言，保证程序不会泄露造成隐私数据的泄漏。
- 2019 扬子毅：高阶逻辑编程语言
  - 将高阶逻辑引入Prolog
- 2020 谢睿峰 董子宁：可验证的函数式程序设计语言
  - 在Lambda Calculus上加入前后条件，同时在编译时调用SMT Solver来证明性质成立

# 往年项目举例

- 2017 沈琦、郝有峰、周昊宇。绘图编程语言
  - 描述各种形状和递归组合
  - 描述各种约束：不可重叠、不可超出边界等
  - 自动检查约束
- 2016 扬子岳。作曲语言
  - 描述乐曲，通过条件循环等方式支持复杂的重用
  - 通过类型系统检查和弦
- 2014 徐泽骅、刘晨昊、包新启。网络协议编程语言
  - 三名网络所的同学设计的描述网络协议的编程语言
  - 设计语法描述越界检查、大小端检查、字段定义等问题
- 2019 张宇博 吴瑾昭 许涵：类型安全的Latex
  - Latex的宏不是类型安全的，设计了一个新版宏，可以编译成普通Latex