

编程语言的设计原理 课程项目

胡振江、赵海燕、熊英飞

2014

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：2~3人为一个小组完成
- 考核时间点：
 - 3月23日：报组队名单给助教
 - 4月16日：选题报告。每组5分钟，介绍选题的意义和目标
 - 5月27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。
 - 5月28日、6月4日：口头报告。每小组30~40分钟，具体时间根据分组情况决定。

课程项目示例（不限于以下题目）

- 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