

编译总结感想

虽然之前参加了比赛，自己也实现过简单的解释器，但是完成实验并没有想象中的轻松，前端虽然没有什么问题（错误处理也遇到了困难），但是后端由于没有写过（比赛负责前端和中端优化，解释器没有复杂的目标代码生成），在完成图着色、SSA解构时遇到了一些困难。而且实验是独立完成，编码压力还是很大的，最终完成一个编译器感觉收获颇丰。在实验过程中，也理解了比赛时中端优化与后端对接不当导致的性能不佳的原因，编译器是一个整体，顾此失彼将不能达到好的效果。

先设计再编码是非常重要的，虽然我实现的编译器继承了比赛的架构，没有遇到太大的问题，但是我了解到一些同学由于是从头设计，可能由于架构原因，在优化过程中发现不好加进去，需要修改的点分散在四处，debug非常艰难。建议在编码前先参考其他人的架构（比如Ayame）。

编译实验代码生成也可以选择LLVM和pcode，但是没有性能分。其实感觉可以在LLVM和pcode上加上实现一些额外功能的给分点，提高其分数上限，比如LLVM IR实现SSA形式（Mem2Reg）可以加一些分数，或者要求pcode可以实现如嵌套函数和词法闭包，垃圾回收等功能，并要求pcode扩展一定规定的语法（如实现类和对象）以获得加分（往解释器方向扩展）。虽然理论课可能没有深入讲这些主题，但是可以给出一些参考资料来帮助实现。

写编译器的日子（写半年了都）大概终于要告一段落了，希望最终编译实验能画上一个圆满的句号。