Rust 核心语言机制的编译验证方法

CCF-华为胡杨林基金形式化专项

研究背景

Rust 编译器的正确性

Rust 语言通过引入所有权机制排除内存安全问题，其依赖于编译器的正确性保障。

Safe Rust 和 Unsafe Rust 的交互

如何让 Safe Rust 与 Unsafe Rust 正确交互，保证完整程序编译和链接后的正确性，仍是未解难题。

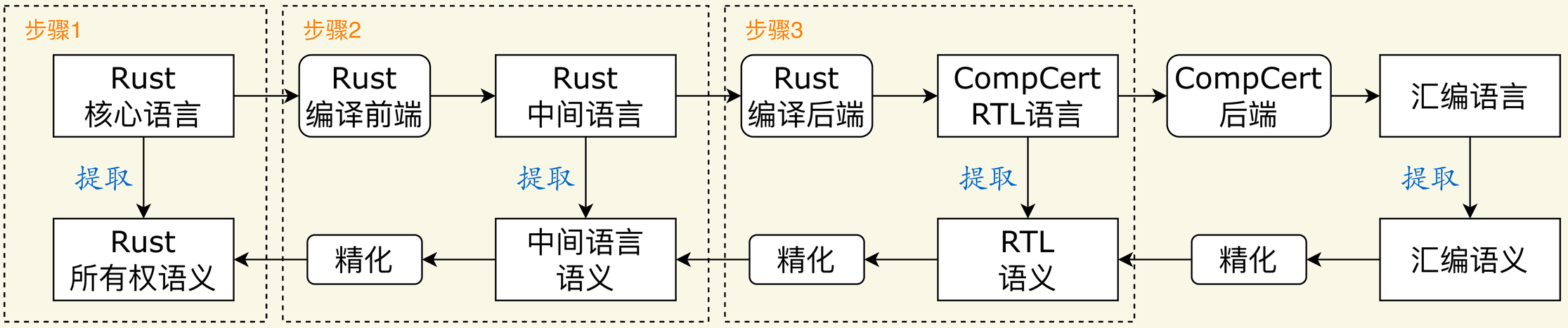
研究目标

提出新型的编译器验证方法，支持Rust 核心语言编译验证 以及Safe/Unsafe Rust 的分离编译验证。

研究方案

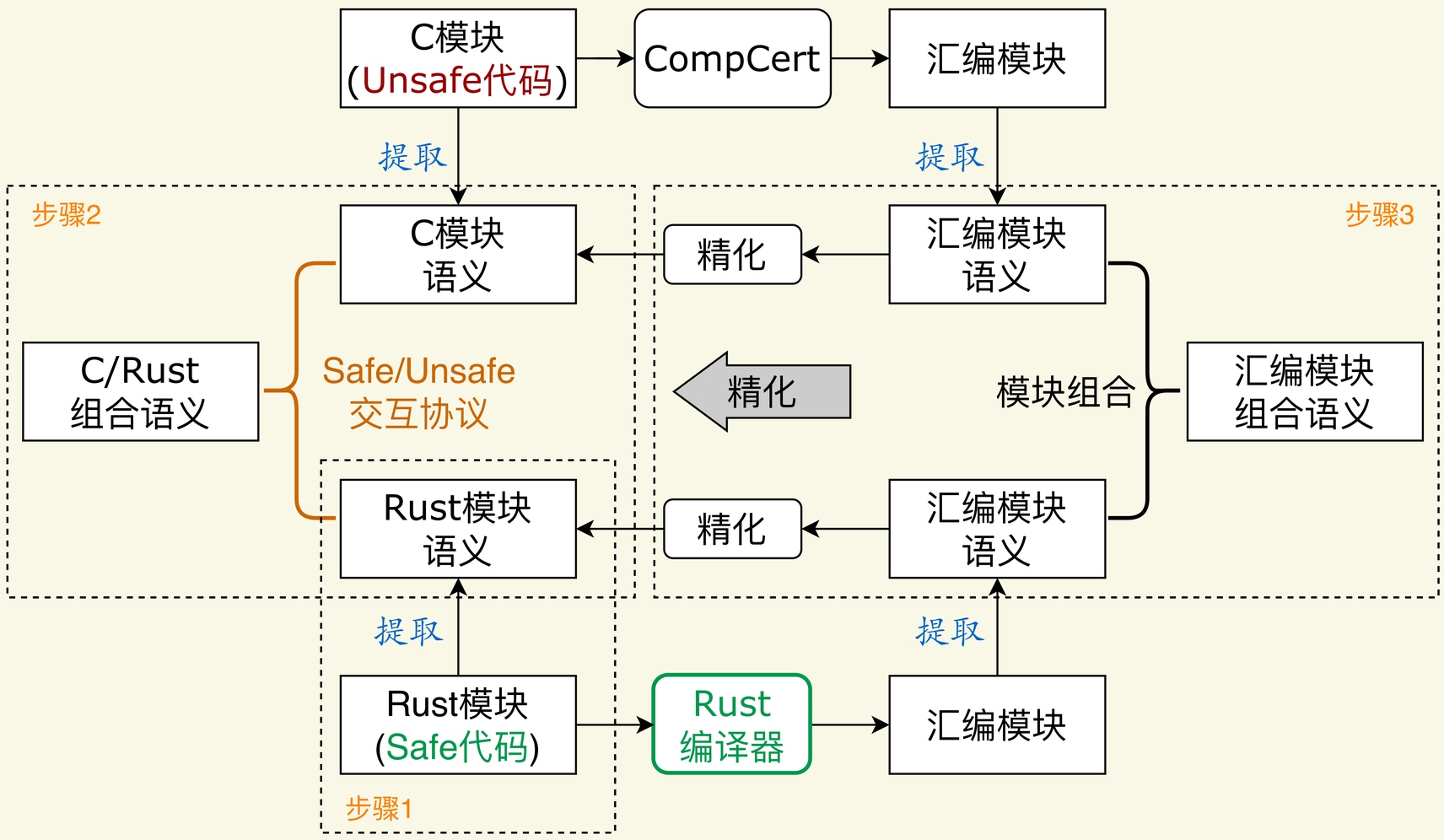
Rust 核心语言编译验证

1. 设计带所有权机制的 Rust 的核心语言。
2. 利用所有权机制生成带自动内存管理的中间语言，并验证该过程。
3. 将中间语言编译到 CompCert 的中间表示并验证，与 CompCert 的编译正确性结合。



Safe/Unsafe Rust 的分离编译验证

1. 设计 Rust 核心语言的开放语义和语言接口。
2. 设计带所有权的语义接口，作为 Safe Rust 和 Unsafe Rust 的交互协议。
3. 将该语义接口与编译器的语义接口整合，获得支持分离编译的语义接口。



汪宇霆　上海交通大学约翰·霍普克罗夫特计算机科学中心

yuting.wang@sjtu.edu.cn

个人主页： https://jhc.sjtu.edu.cn/~yutingwang/