**背景切入和研究动机: 为什么研究?**

零知识证明的重要性越来越大,他在密码学, 区块链,多方计算等方面越来越得到重视. 零知识证明首先是作为隐私问题的解决方案出现的，但它们最近也被视为为可扩展性问题的完美解决方案。因此，这些加密证明已经成为对区块链社区非常有吸引力的工具，而且相应的算法已经在很多应用程序中得到了部署与整合。

在零知识证明的底层工具链中, CIRCOM语言到R1CS这一步非常重要, zk-SNARK不能直接应用于任何计算问题, 而转换成R1CS就是将一些计算问题转换为正确形式的第一步.

但是在这一段地方circom存在着很多局限, 首要问题就是拓展性较差, A与B合并后所生成的R1CS与A和B独立生成的R1CS在形式上毫无关联, 而这与R1CS本身表达能力局限性有关外,根本原因便是等价的程序本身可以生成无数个等价的R1CS, 所以我们需要在R1CS生成时提出一个范式,这对我们验证程序的等价性, 正确性, 包括后续更加深入研究R1CS的可合并性方面都将大有裨益.

**为什么会出现这个问题**

1. R1CS不同规则的合并, 中间变量的选择

**应该怎么做:**

等任务书

**目前做了什么:**

把circom和R1cs视为编译前后的两种语言, 这个问题其实近似于编译语义一致性的研究

读了一些相关的论文和专利书, 发现相关研究基本是从数据流和语法树两个方向出发. 目前先以数据流为基础

目前以提出基本方案, 形成数据流->根据数据流找出关键的节点->一这些节点为基础进行R1CS的生成与合并

这样得到的R1CS是从程序中的数据流和对变量的计算操作出发, 与具体的程序实现细节是解耦的

目前在一些简单的程序上尝试进行该方法, 发现可以解决一些由于变量定义, 计算顺序混乱导致R1CS形式上的不同, 而这也是我之前调研目前主流CIRCOM编译器时发现的主要问题

**未来打算:**

等任务书