1. 各位老师，同学好，今天由我来给大家做一个开题报告，我的论文题目是，基于数据流的 R1CS 等价性检查与范式生成

2. 这是我今天报告的一个目录

3. 首先我来介绍一下这个课题的研究目的与意义

4. 零知识证明在现代社会中越来越体现出他的重要性，越来越多的加密社区正在寻求通过零知识证明技术来解决一些区块链的难题，其中两个最主要的应用便是隐私安全和可扩展性问题。无论是从用户角度还是从开发和角度，对信息隐私安全性的加剧都使零知识证明在隐私方面的优势越来越得到重视。同时，零知识证明在元宇宙和 web3， 加密货币选择，金融行业，隐私保护等领域也在发挥着越来越重要的作用

5. 但是零知识证明并不能直接应用于任何计算问题，相反，我们必须将问题转换为操作的正确形式。这种形式被称为“二次算数程序（QAP）”，具体到一次零知识证明的过程中，也就是右边这张图所展示的零知识证明的底层工具原理，我们首先是把问题转换成 Circom 语言，再从 Circom 中提炼出 R1CS 的约束，再由约束转换到 QAP 的形式。在此之后，还有另一个相当复杂的过程来为这个 QAP 创建实际的“零知识证明”，还有一个单独的过程验证所收到的证据，

   在其中，由 Circom 到 R1CS 约束这一步转换存在着很多局限

6. 通过我之前进行的一些调研以及对当前主流编译器的一些实际操作，可以得出局限性主要来自于两个原因，首先是 R1CS 的可合并性较差，比如说由程序片段 A 生成的 R1CS 约束组 a，由程序片段 B 生成约束组 b，那么当直接由程序 A+B 生成的 R1CS 约束组，可能从形式上看与 a 和 b 毫无关联。

   还有一个原因就是即使是相同的算术电路，它所对应的 R1CS 约束组的形式也是多种多样的。比如我们看这个例子，上面是规规矩矩经过引入中间变量，拍平等一系列操作后得到的约束组，下面是 github 上一个当前主流的 circom 编译器编译得到的约束组，可以看到即使是同一个 circom 程序所对应的 R1CS 约束组，但从形式上看也很难判定他们代表的电路是否是等价的,而实际上,这两组约束都是合法的。

   所以，如果我们能在众多等价的 R1CS 约束组中提出他们的范式，那么我们就可以比较容易地判断他们的等价性和正确性， 这对我们验证程序的等价性和正确性，包括后续更加深入研究 R1CS 的可合并性等方面都将大有裨益。

7. 接下来我要先介绍一下我的参考文献，顺带解说一下目前国内外相关研究的现状。

8. 将 Circom 和 R1CS 视为编译前后的两种语言，R1CS 范式生成问题的研究其实更加近似于编译语义一致性的研究。目前国内外也有一部分专利和论文在其他语言的编译范式生成上提出了思路和解决方法。这些相关研究基本上从数据流[2]、语法树[3]或者语义映射[4]这几个方面出发，这也是我这一页参考文献的主要内容

9. 而基于数据流的分析，根据我检索到的参考文献来看，国内外关于数据流图设计[5]、数据流图分析[6]、由程序语言生成数据流图的算法[7]以及数据流图一致性检查[8]等方向均有较成熟的研究。可见在数据流与程序分析这个领域，前人的经验十分充分，对后续研究的开展将提供很多理论基础。

10. 接下来介绍一下预期研究成果

11. 基本是分成四个小目标，也差不多是我研究的四个阶段性目标
    1. 较为深入的分析 Circom 编译器生成 R1CS 约束的规则，并详细的总结等价 R1CS 约束在形式上不同之处的产生规律。
    2. 设计表达 R1CS 约束中变量逻辑关系的数据流图形式
    3. 制定由数据流图生成 R1CS 范式的生成规则，实现任意 R1CS 约束到其范式的转换过程
    4. 以数据流图为基础，设计并实现对 R1CS 范式生成以及等价性对比的算法，使得对于输入的任意等价 R1CS，均能输出形式完全一致的 R1CS 约束范式。

12. 然后看一下目前所做的一些工作

13. 首先是看了相关的论文和专利书，基本敲定以数据流为基础，完成 R1CS 范式的生成

14. 提出目前的基本方案，形成数据流，图中的关键节点，以节点为基础生成 R1CS 约束，这样的话所生成的 R1CS 是从程序中的数据流以及变量之间的逻辑关系为基础，与编译器内部的逻辑和程序实现的细节达到一个解耦的状态，然后手动对一些比较简单的约束组进行了试验，发现基本能满足一致的要求，解决了一些诸如中间变量的选择，变量定义的顺序这种比较明显又简单的问题所引起的 R1CS 约束组上的不同

15. 最后是我的计划进度安排

16. 我主要也是根据之前的各阶段研究成果层层递进的，

17. 那么我的报告到此结束，感谢各位老师和同学的倾听，谢谢大家。