# 基于数据流分析的R1CS语言等价性验证及范式生成

答辩人: 施宸昊

指导老师: 李国强

2023年5月

饮 水 思 源 • 爱 国 荣 校

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

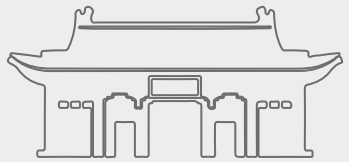
# 目录

01 研究背景

02 算法设计

03 实验分析

04 总结与展望



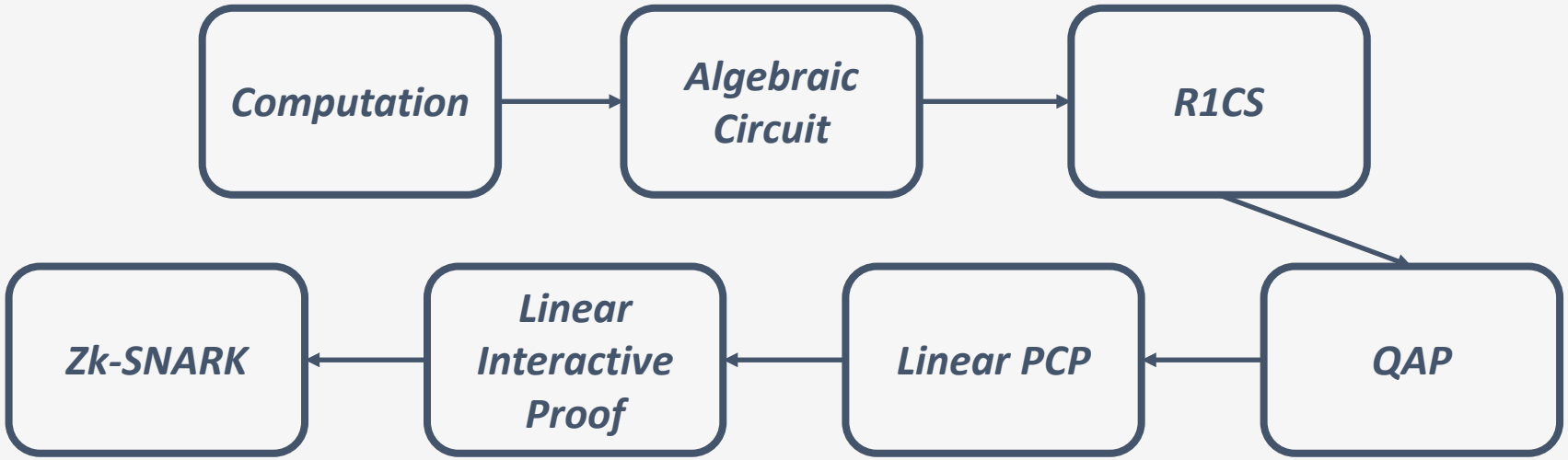
# 研究背景

上海交通大學 SHANGHAI JIAO TONG UNIVERSITY

## 零知识证明在现代社会中正在变得越来越重要，应用越来越广泛。

- **两个主要应用:** 隐私安全、可拓展性
- **关键应用领域:** 元宇宙与Web3、加密货币选择、金融行业、隐私保护

## 零知识证明的应用需要复杂的转换过程



# 研究背景

上海交通大学 SHANGHAI JIAO TONG UNIVERSITY

- **在算术电路到R1CS的转换步骤上存在局限性**
- **可合并性差**
  - 程序$A$生成约束$a$，程序$B$生成约束$b$
  - 程序$A+B$生成的约束$c$可能与$a$与$b$在形式上毫无关联
- **等价算术电路转换出的R1CS形式多样**
  - 以$x^3+x+5=out$为例:

| A | B | C |
|---|---|---|
| [0, 1, 0, 0, 0, 0] | [0, 1, 0, 0, 0, 0] | [0, 0, 0, 1, 0, 0] |
| [0, 0, 0, 1, 0, 0] | [0, 1, 0, 0, 0, 0] | [0, 0, 0, 0, 1, 0] |
| [0, 1, 0, 0, 1, 0] | [1, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 1] |
| [5, 0, 0, 0, 0, 1] | [1, 0, 0, 0, 0, 0] | [0, 0, 1, 0, 0, 0] |

| A | B | C |
|---|---|---|
| [0, 1, 0, 0] | [0, 1, 0, 0] | [0, 0, 0, 1] |
| [0, 0, 0, 1] | [0, 1, 0, 0] | [5, 1, 1, 0] |

**两组R1CS约束均合法**

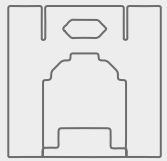
SJTU

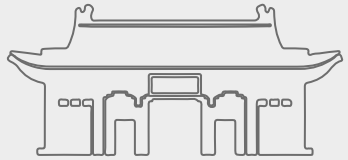
01 研究背景

02 算法设计

03 实验分析

04 总结与展望





# 算法设计：一对等价约束组

- 以一对等价约束组在算法中的各个步骤的中间输出介绍算法的整体流程

- 等价约束组:

| A | B | C |
|---|---|---|
| [0, 1, 0, 0, 0, 0] | [0, 1, 0, 0, 0, 0] | [0, 0, 0, 1, 0, 0] |
| [0, 0, 0, 1, 0, 0] | [0, 1, 0, 0, 0, 0] | [0, 0, 0, 0, 1, 0] |
| [0, 1, 0, 0, 1, 0] | [1, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 1] |
| [5, 0, 0, 0, 0, 1] | [1, 0, 0, 0, 0, 0] | [0, 0, 1, 0, 0, 0] |

$Variable\ mapping = (\sim one, x, , \sim out, x^2, x^3, sym_1)$

| A | B | C |
|---|---|---|
| [0, 1, 0, 0] | [0, 1, 0, 0] | [0, 0, 0, 1] |
| [0, 0, 0, 1] | [0, 1, 0, 0] | [5, 1, 1, 0] |

$Variable\ mapping = (\sim one, x, \sim out, x^2)$

**电路约束**

$$\begin{cases} x \times x = x^2 \\ x \times x^2 = x^3 \\ x + x^3 = sym_1 \\ sym_1 + 5 = out \end{cases}$$

# 算法设计：数据流图的建立

上海交通大学 SHANGHAI JIAO TONG UNIVERSITY

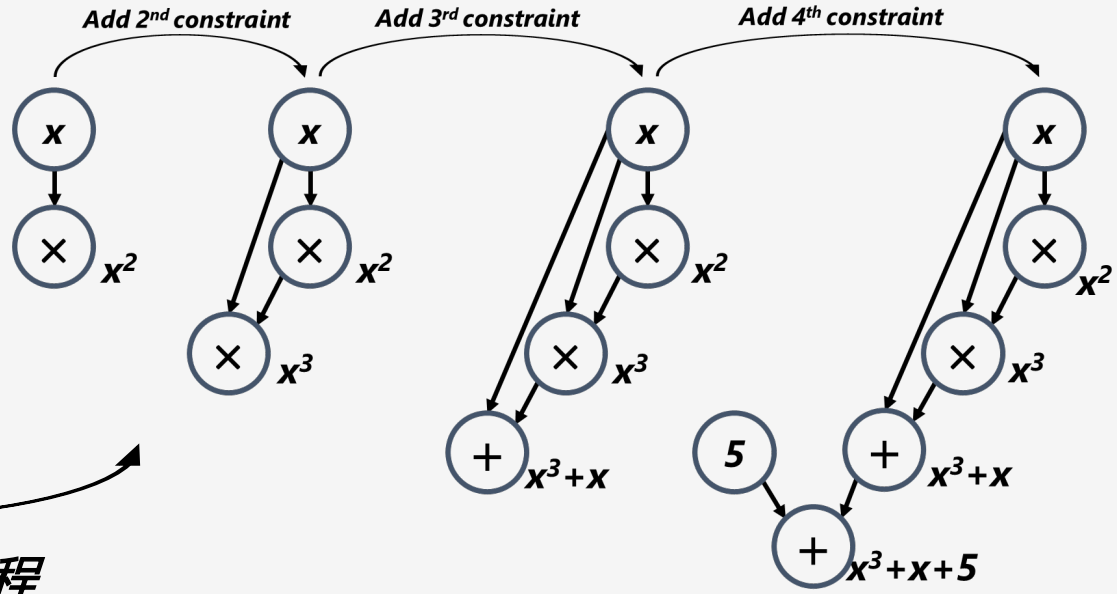
## 数据流图的建立主要分两个步骤:

- 按照R1CS所满足的等式，$(\mathrm{s} \times \mathrm{a}) * (\mathrm{s} \times b) = \mathrm{s} \times c$，将每一个约束转化为算式

$$(0 \quad 1 \quad 0 \quad 0) \times (0 \quad 1 \quad 0 \quad 0) = (0 \quad 0 \quad 0 \quad 1) \rightarrow \mathrm{x}_2 * \mathrm{x}_2 = \mathrm{x}_4$$

- 合并算式, 得到DAG

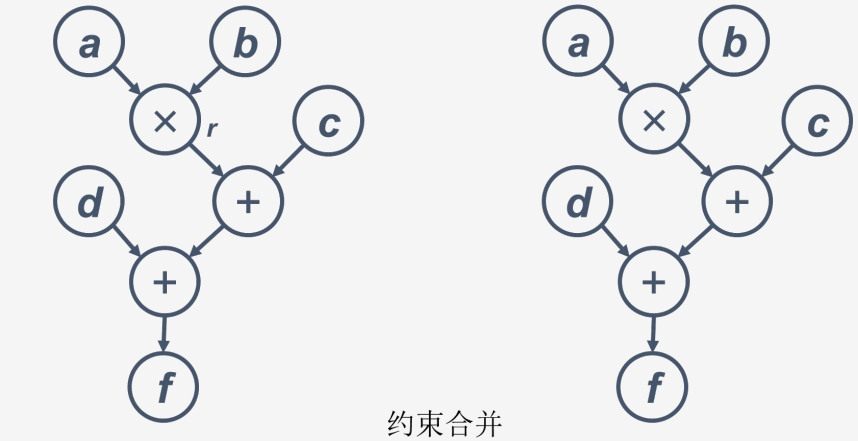
| A | B | C |
|---|---|---|
| [0, 1, 0, 0, 0, 0] | [0, 1, 0, 0, 0, 0] | [0, 0, 0, 1, 0, 0] |
| [0, 0, 0, 1, 0, 0] | [0, 1, 0, 0, 0, 0] | [0, 0, 0, 0, 1, 0] |
| [0, 1, 0, 0, 1, 0] | [1, 0, 0, 0, 0, 0] | [0, 0, 0, 0, 0, 1] |
| [5, 0, 0, 0, 0, 1] | [1, 0, 0, 0, 0, 0] | [0, 0, 1, 0, 0, 0] |




数据流图建立过程

# 算法设计：数据流图的建立

## 将约束的组合与拆分的影响降到最低

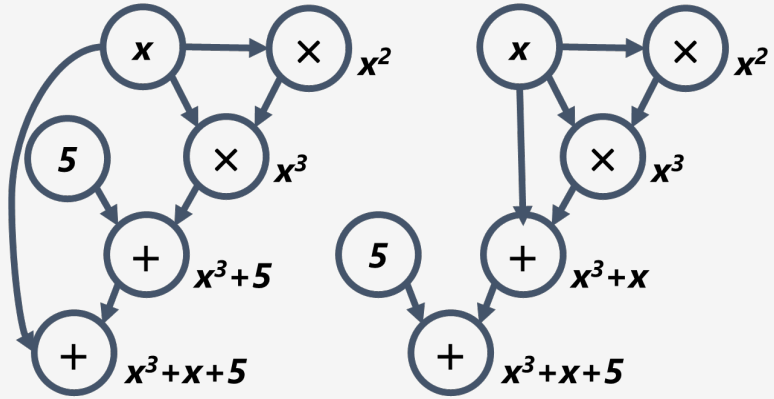



**所带来的影响：**

- 影响中间变量的选择
- 影响约束的数量与形式
- 影响变量映射关系

**建立中间变量，维持总体结构不变**

## 数据流图之间仍然存在不同




**在数据流图中，连续加法执行顺序可能不同**

- 约束组A：$(x^3 + x) + 5 = out$
- 约束组B：$x^3 + (x + 5) = out$

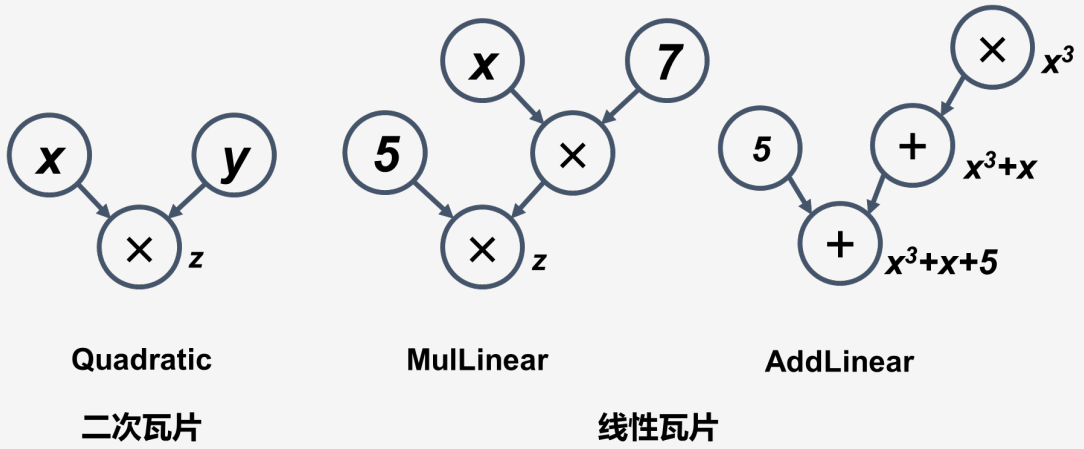
# 算法设计：瓦片生成

## 总体思路

- 参考编译原理中IR-Tree到汇编语言语法树的转换中瓦片选取
- 自定义瓦片类型
- 以合法瓦片类型分割数据流图

## 瓦片的三种类型

- **二次瓦片：** 如$x * y = z$
- **线性瓦片：**
  - **MulLinear：** 如$(7 * x) * 5 = z$
  - **AddLinear：** 如$x^3 + x + 5 = z$

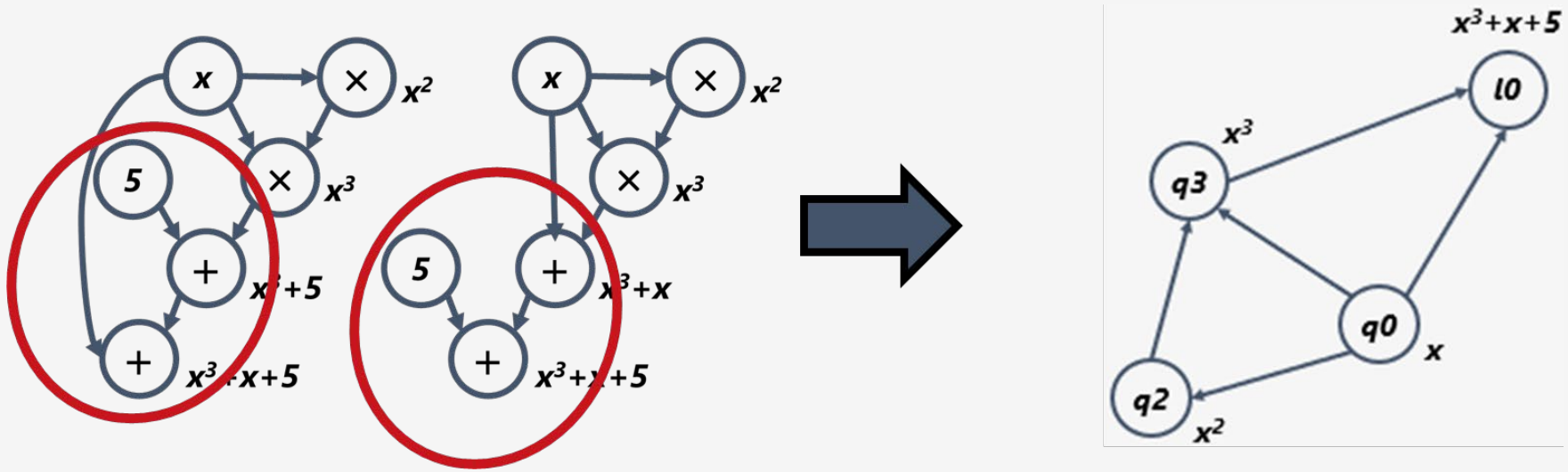



## 三个考量:

- 将约束合并步骤暂时搁置
- 以未进行约束合并的范式作为基础
- 瓦片选取的算法实现也相对简单

# 算法设计：数据流图进一步抽象

- 将选出的线性瓦片抽象成一个大的节点，二次瓦片中的点保持不变




- 边的类型有以下几种
  - **RNode 节点到 RNode 节点**：与抽象前的数据流图保持一致
  - **RNode 节点到线性瓦片抽象节点**：当且仅当抽象节点所代表的线性瓦片中存在该RNode节点时存在
  - **线性瓦片抽象节点到线性瓦片抽象节点**：当且仅当两个抽象节点所代表的线性瓦片存在公有的 RNode 节点时存在

# 算法设计：瓦片权重计算



## 使用Weighted PageRank算法

- Xing W, Ghorbani A. Weighted pagerank algorithm[C]//Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004. IEEE, 2004: 305-314.

## 权重计算公式

- $PR(u) = (1-d) + d \sum_{v \in B(u)} PR(v) W_{(v,u)}^{in} W_{(v,u)}^{out}$
- $W_{(v,u)}^{in} W_{(v,u)}^{out}$是根据节点v及其邻居的入度和出度计算出的权重

## 对线性瓦片,使用其约束中系数的方差作为权重

- 降低抽象后数据流图的对称性
- 消除瓦片之间的相同权重

# 算法设计：对线性瓦片的调整

## 调整范围：在线性瓦片中新引入的变量

- 将线性瓦片抽象成一个节点之后，数据流图中丢失了线性瓦片具体结构的信息
- 对于线性瓦片中新增的节点，没有有效的排序标准

## 提出排序标准

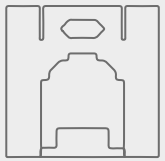
- $weight = \sum_{other\ linear\ tiles} | field * weight\ of\ linear\ tile |$

## 原因

- 以变量在约束中的出现情况反映变量重要性
- 保持只在单一线性约束中出现的变量权重相同

01

研究背景



02

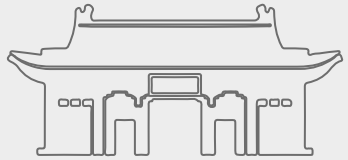
算法设计

03

实验分析

04

总结与展望



# 实验分析

## 数据集设计

- 总结等价R1CS约束组生成规律，对数据集分类
- 在每个类别中：


## 测试结果展示

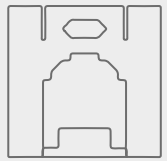
| 等价约束组生成原因 | 实验组数 | 成功数 | 通过率 |
|---|---|---|---|
| R1CS中变量顺序的替换。 | 55 | 55 | 100% |
| R1CS中约束顺序的变换。 | 21 | 21 | 100% |
| R1CS中单个线性约束中多个新变量的引入。 | 15 | 15 | 100% |
| R1CS中多个线性约束中多个新变量的引入且存在新变量共用。 | 15 | 15 | 100% |
| R1CS中约束的合并与拆分。 | 6 | 6 | 100% |

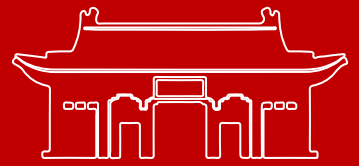
01 研究背景

02 算法设计

03 实验分析

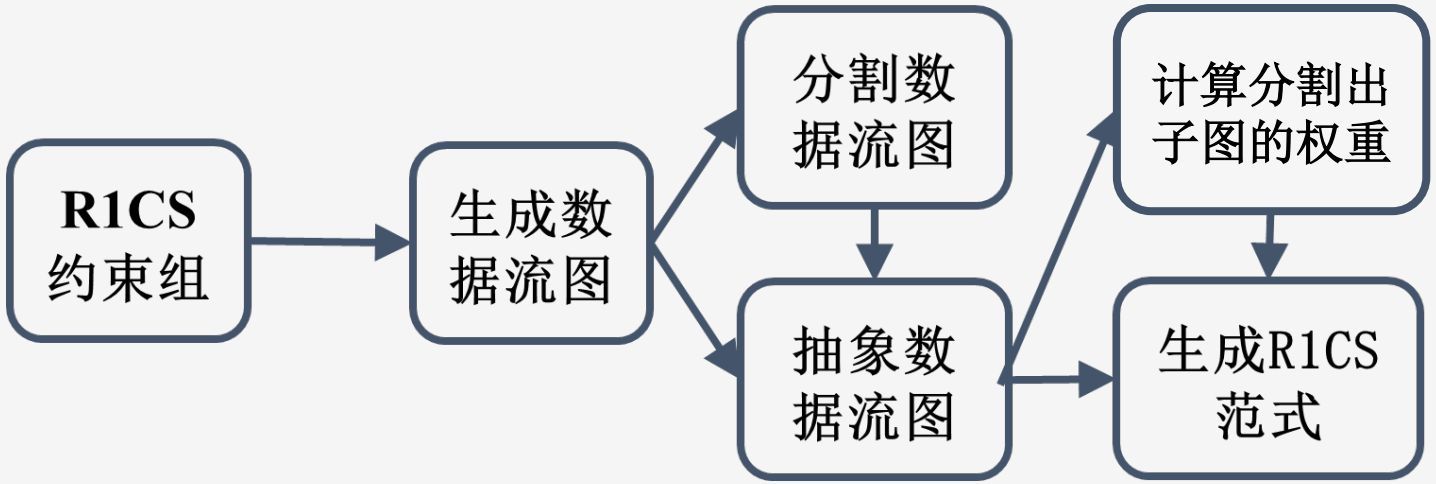
04 总结与展望





# 本文贡献

- 提出R1CS约束组范式的形式并设计范式生成算法




- 总结R1CS等价约束组生成的规律
- 设计R1CS等价约束组的测试集

# 未来展望

上海交通大学 SHANGHAI JIAO TONG UNIVERSITY

## 约束之间的合并规则

- 在瓦片分割的步骤中，将二次约束的瓦片的形式限制在了最简单的形式
- 生成的约束组中的矩阵过于稀疏

## 更加完备的测试集

- 当前领域内并不存在公开的较为完备的数据集
- 本论文所提出的数据集仅基于主流的Circom编译器的约束生成逻辑

## 更有效率的算法流程

- 数据流图的生成、瓦片的划分等步骤存在可并行的部分
- 算法运行时内存占用较大
- 对数据流图提出范式，将等价的R1CS约束组产生的数据流图归约到相同的形式

上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

# 感谢各位评审老师

饮 水 思 源　爱 国 荣 校