以两个等价的约束组在目前所实现的流程中的变化来介绍目前的研究成果

约束组 A:

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

约束组 B:

A	B	C
0,1,0,0,	0,1,0,0	0,0,0,1
0,1,0,0	0,0,0,1	-5,-1,1,0

1. 建立数据流 DAG,
   DAG 中节点的数据结构设计如下:

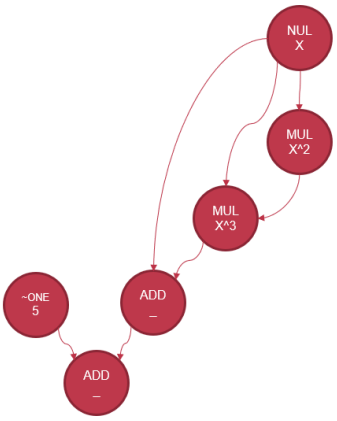
```
def __init__(self, id, op=Op.NULL, name="_", con=0):  # 初始化类的属性
    self.id = id
    self.op = op
    self.const = con
    self.name = name
    self.child = []
    self.father = []
```

在该数据结构中，变量和操作符使用同一种节点类型，以确保当 R1CS 二次约束与线性约束合并或拆分时，不会对 DAG 的结构产生影响

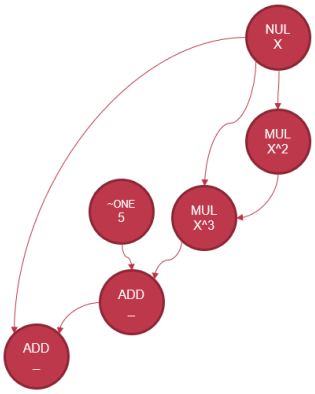
- id 用以辨识节点,
- op 存在 NULL, MUL, ADD 三种枚举类型，用以标明该节点是如何由其父亲节点计算得出,
- const 属性为"~one"变量专用，用以表示常数项的大小，同时,R1CS 的约束中,如果某变量前的系数为非 1 的常数,在树中也会~one * 该变量来表示. 在树中每次使用~one 变量均会新创建一个 name 属性为~one 的节点

A 与 B 两个约束组在 DAG 完成后的结构如下:

约束组 A:



约束组 B



可以发现，两个 DAG 结构上的不同源于连加中相加的顺序不同，例如约束组 A 中先加 x 再加常数项，而约束组 B 中先加常数项再加 x.

虽然形如 pagerank 之类的算法对图的微小变化不太反应明显，同时也试了一些等价约束组形成的 DAG，发现对于同一节点，权重的波动值大概是权重本身的 10%，在大部分情况下不会对节点的权重排序产生影响，但是一旦数据流 DAG 中权重的序列发生变化，范式的生成将难以进行,因此要对 DAG 进行进一步的抽象.

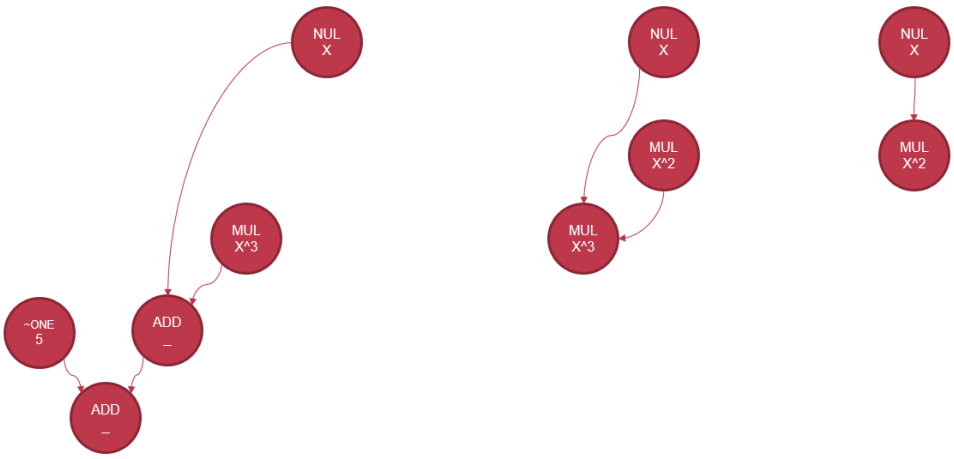
如果把 DAG 中的连加链与最简单的二次型单元，也就是形如 A*B=C 这样的单变量乘法，看作组成 DAG 的两个单元类型(分别对应 circom 编译器中的 linear 和最简单的 quadratic 类型约束)，那么连加中相加顺序的不同就会变成该单元内部的问题，而单元之间的关系并不会因为连加顺序的不同造成影响，具体情况在瓦片选择中进行说明

2. 瓦片选择

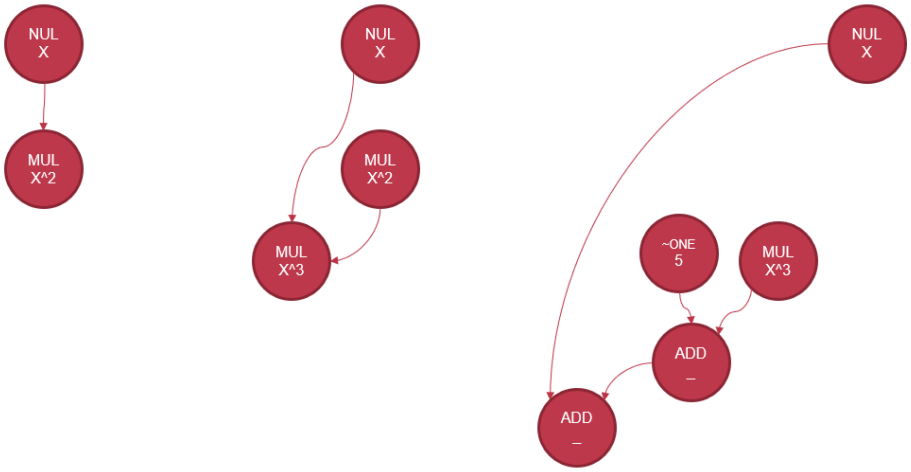
瓦片选择时,我们将上一个步骤中的树,分割成连加链与最简单的二次型单元，也就是形如 A*B=C 这样的单变量乘法，这样选取有几个考量:

1. 将约束合并步骤暂时搁置，待后续步骤获取树中的更多信息后在进行
2. 产生未合并的范式后，如有产生合并范式的需求,只需在未合并的范式上应用固定算法即可，相对简单
3. 瓦片选取的算法实现也相对简单

根据上一步骤中的基本单元对树进行瓦片选取获得瓦片如下:
约束组 A:

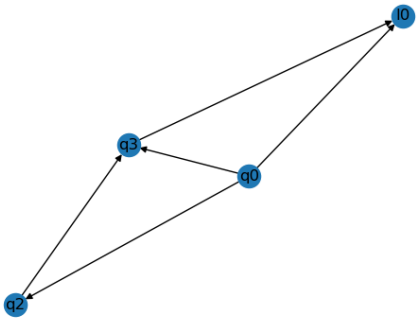


约束组 B:



可以发现连加链的顺序不同不会影响到瓦片的选取,

然后我们再将连加链抽象成一个大的节点,而剩下的 quadratic 的瓦片保持不变, quadratic 的瓦片中的节点到 linear 的瓦片中节点的边均转变为到 linear 抽象节点的边. 这是约束组 A 和约束组 B 抽象出的 DAG 为:



其中 l0 时 linear 瓦片抽象出的节点，而 q0,q2,q3 是 quadratic 瓦片中的节点.至此，约束组中因为约束合并和行列变换引起的不同被完全消除

3. 瓦片权重计算:
   首先对上一个步骤所得到的图应用 PageRank 算法，得到各节点的权重,
   然后以此计算瓦片的权重,其中 quadratic 瓦片的权重为各节点的算术平均值, linear 瓦片的权重即为其抽象节点的权重
4. 约束生成
   以瓦片为单位,以权重的降序生成约束，先生成 quadratic 约束，再生成 linear 约束. 约束组 AB 被转化为:
   [~one, x, x^2, x^3, ~out]

A:	B:	C:
0, 1, 0, 0, 0	0, 0, 1, 0, 0	0, 0, 0, 1, 0
0, 1, 0, 0, 0	0, 1, 0, 0, 0	0, 0, 1, 0, 0
5, 1, 0, 1, -1	1, 0, 0, 0, 0	0, 0, 0, 0, 0