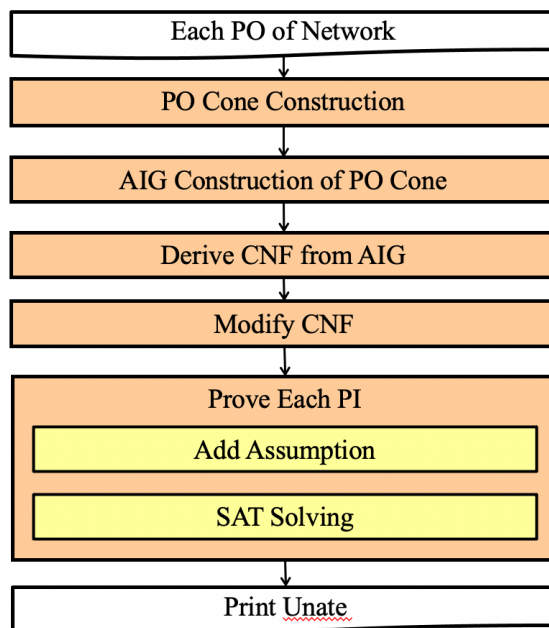


## 1. Approach

### (1) Flow

We extract the cone of certain PO to construct the corresponding AIG and CNF. Then we add CNF to SAT solver, and the CNF we add to SAT solver is  $F \wedge \neg F \wedge (a_i \vee x_i \vee \neg x'_i) \wedge (a_i \vee x'_i \vee \neg x_i)$ , where  $x_i$  is the PI of  $F$  and  $x'_i$  is the PI of  $\neg F$ . To prove  $x_i$  is positive unate, we need to make assumption that  $a_i = 1, x_i = 0, x'_i = 1$  while  $a_j = 0, j \neq i$ , and the result must be UNSAT. To prove  $x_i$  is negative unate, we just need change the assumption to  $x_i = 1, x'_i = 0$ . Therefore, we only need to construct CNF for each PO once.



### (2) Data structure

I construct a class Network to store the information of the current network. It contains the pointer of original network, the AIG pointer of original network, the pointer of current PO cone.

Besides, we also maintain a map of PI name to proving result to have correct result while proving cones.

### (3) Optimization

I try four optimization command to reduce the size of AIG : DC2, drw, dfraig, balance.

Besides, I have use a alternative SAT solver GLUCOSE to compare with MINISAT.

## 2. Result

We tried four different approach to solve the problem. The approaches are list

below and the experimental result is shown as the table. There are total 13 cases, and the table shows the runtime to solve each case by different approaches.

Approach 1 : Glucose without optimization

Approach 2 : Glucose with ten iteration optimization

Approach 3 : Minisat without optimization

Approach 4 : Minisat with one iteration optimization

Approach 5 : Minisat with ten iteration optimization

	Approach 1	Approach 2	Approach 3	Approach 4
case1	6.48	24.36	4.91	4.93
case2	0.02	0.53	0.02	0.02
case3	0.01	0.42	0.03	0.03
case4	0.05	4.37	0.02	0.19
case5	0.07	2.87	0.12	0.12
case6	0.01	0.22	0.02	0.01
case7	20.5	96.41	50.47	50.18
case8	0.08	0.55	0.08	0.08
case9	0.02	0.23	0.02	0.02
case10	6.33	19.81	3.34	3.63
case11	2.02	22.36	1.89	2.08
case12	85.56	232.61	51.08	51.26
case13	10.94	74.16	10.38	10.42
Ave	10.1607692	36.8384615	9.41384615	9.45923076

#### Addition Test Case

	Approach 5	Approach 4
div.aig	x	x
hyp.aig	x	x
log2.aig	12212.19	10583.66
multiplier.aig	x	x
sqrt.aig	x	x
square.aig	1258.91	681.93
voter.aig	x	x

### 3. Analysis

Both approach 3 and approach 4 use minisat as the SAT solver. Compared the

runtime of the two approaches, the results show that optimization only has little improvement to case 7 while doing optimization will increase the runtime in all the other cases.

Approach 1 and approach 3 are only differ in SAT solver. The result shows that although approach 1 performs better in case 7, the runtime of approach 3 is faster than the runtime of approach 1.

For a longer runtime limit, I can solve two more test case. I have tried two different way, the result can show that approach four can solve the case faster.