

# Logic Synthesis

## 1 Description of the problem

In this project, our goal is to solve the *NOR Logic Synthesis Problem* (NLSP): given a specification of a Boolean function  $f(x_1, \dots, x_n)$  in the form of a truth table, find a NOR-circuit satisfying the specification that minimizes depth (and, in case of a tie in depth, with minimum size). An instance of NLSP consists in:

- $\mathbf{n} :=$  “Number of input signals”
- $\mathbf{y}_t :=$  “Desired output signal, described by row  $t$  in the truth table”, where  $t \in \{0, 1, \dots, 2^n - 1\}$

## 2 Decision variables

Given the number of input signals  $n$ , the depth  $d$ , and the truth table of the logical circuit, I defined the following variables:

- $\mathbf{Z}_{i,j} :=$  “The node  $(i,j)$  contains a constant zero”, where
  - $0 \leq i \leq d$
  - $0 \leq j < 2^i$
- $\mathbf{N}_{i,j} :=$  “The node  $(i,j)$  contains a NOR gate”, where
  - $0 \leq i \leq d$
  - $0 \leq j < 2^i$
- $\mathbf{I}_{i,j,k} :=$  “The node  $(i,j)$  contains the input  $k$ ”, where
  - $0 \leq i \leq d$
  - $0 \leq j < 2^i$
  - $1 \leq k \leq n$
- $\mathbf{B}_{i,j}^{(t)} :=$  “Boolean value of the node  $(i,j)$  for the row  $t$  of the truth table”, where
  - $0 \leq i \leq d$
  - $0 \leq j < 2^i$
  - $0 \leq t < 2^n$

For example, for a NOR-circuit that implements the functionality of an AND gate (see figure 1), with  $n = d = 2$ , one possible solution (variable assignation) is shown in figure 2.

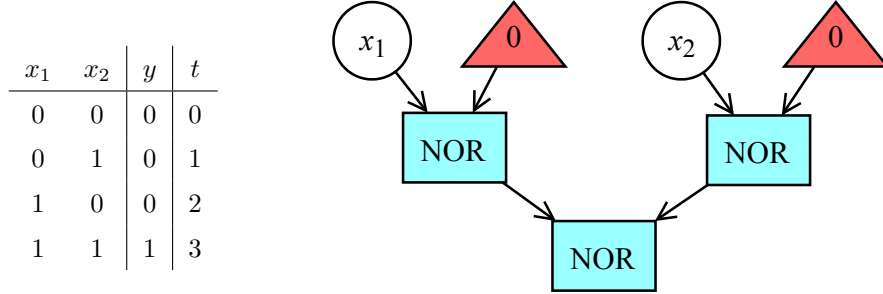


Figure 1: Truth table of  $y = \text{AND}(x_1, x_2)$  and NOR-circuit implementing it.

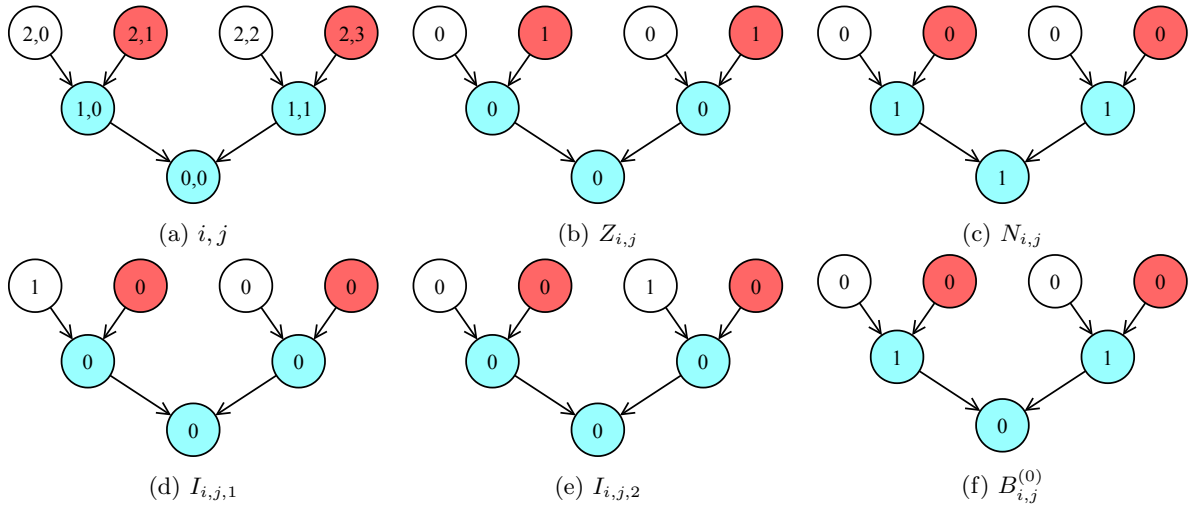


Figure 2: Visual representation of  $(i, j)$  and the variables.

### 3 Constraints

In order to simplify the definition of the constraints, I define three functions. Given the variable  $v_{i,j}$ , with  $v_{i,j} \in \{Z_{i,j}; N_{i,j}; I_{i,j,k}; B_{i,j}^{(t)}\}$ ,

- **left**( $v_{i,j}$ ) := “Variable corresponding to the one on the left of  $v_{i,j}$ ” =  $v_{i+1,2 \times j}$ .
- **right**( $v_{i,j}$ ) := “Variable corresponding to the one on the right of  $v_{i,j}$ ” =  $v_{i+1,2 \times j+1}$ .
- **bit**( $k, t$ ) := “Boolean value of  $x_k$  in the  $t$ -th row of the truth table” = “Value of the position  $k$  of the binary representation of  $t$  (i.e.  $t_k \in \{t_1 t_2 \dots t_n\}$ )”.

So, the constraints are the following:

- The output of the circuit is equal to the desired value for each row  $t$  of the truth table.

$$\begin{aligned} B_{0,0}^{(t)} &= y_t \\ \forall t &< 2^n \end{aligned}$$

- NOR gates are not allowed on the leaves of the circuit.

$$\begin{aligned} N_{d,j} &= 0 \\ \forall j &< 2^d \end{aligned}$$

- Force children (if any) of each node to be 0 if the node is not a NOR gate.

$$\begin{aligned} \mathbf{left}(Z_{i,j}) + \mathbf{right}(Z_{i,j}) &\geq (1 - N_{i,j}) \cdot 2 \\ \forall i &< d \ \forall j < 2^i \end{aligned}$$

- Force non-symmetry of NOR gates' children.

– Do not allow the same input on both sides.

$$\begin{aligned} \mathbf{left}(I_{i,j,k}) + \mathbf{right}(I_{i,j,k}) &\leq 1 \\ \forall i &< d \ \forall j < 2^i \ \forall k \leq n \end{aligned}$$

- Link each NOR gate with its corresponding value, which is the NOR operation between both children.

$$\begin{aligned} (1 - \mathbf{right}(B_{i,j}^{(t)}) - \mathbf{left}(B_{i,j}^{(t)})) - 2 \cdot B_{i,j}^{(t)} &\leq 1 - N_{i,j} \\ (1 - \mathbf{right}(B_{i,j}^{(t)}) - \mathbf{left}(B_{i,j}^{(t)})) - 2 \cdot B_{i,j}^{(t)} &\geq 2 \cdot N_{i,j} - 3 \\ \forall t &< 2^n \ \forall i < d \ \forall j < 2^i \end{aligned}$$

- Link each constant 0 signal with ‘false’.

$$\begin{aligned} 1 - Z_{i,j} &\geq B_{i,j}^{(t)} \\ \forall t &< 2^n \ \forall i \leq d \ \forall j < 2^i \end{aligned}$$

- Link each input signal that has value 1 in the truth table, with ‘true’.

$$\begin{aligned} 1 - I_{i,j,k} &\geq B_{i,j}^{(t)} - \mathbf{bit}(k, t) \\ 1 - I_{i,j,k} &\geq \mathbf{bit}(k, t) - B_{i,j}^{(t)} \\ \forall 1 &\leq k \leq n \ \forall t < 2^n \ \forall i \leq d \ \forall j < 2^i \end{aligned}$$

- Force each node to be only of one type.

$$Z_{i,j} + N_{i,j} + \sum_{k=1}^n I_{i,j,k} = 1$$

$$\forall i < d \ \forall j < 2^i$$

### 3.1 Worsen performance

We tried to use the same constraints than in CP, but the following ones affected negatively to the performance of the program.

- Force non-symmetry of NOR gates' children.
  - Allow an input on the left only if there is an input on the right.

$$\mathbf{left}(I_{i,j,k}) + \mathbf{right}(Z_{i,j}) + \mathbf{right}(N_{i,j}) \leq 1$$

$$\forall i < d \ \forall j < 2^i$$

- Allow a constant zero on the left only if there is not a NOR gate on the right.

$$\mathbf{left}(Z_{i,j}) + \mathbf{right}(N_{i,j}) \leq 1$$

$$\forall i < d \ \forall j < 2^i$$

- Inputs on the left must be smaller than the ones on the right.

$$\mathbf{left}(I_{i,j,k}) + \mathbf{right}(I_{i,j,l}) \leq 1$$

$$\forall i < d \ \forall j < 2^i \ \forall k, l < n \text{ with } k < l$$

## 4 Minimize function

We want to **minimize** the size of the circuit:

$$\sum_{i=0}^{d-1} \sum_{j=0}^{2^i-1} N_{i,j}$$

## 5 Extra comments

Without using any **timeout**, we were able to solve all the problems correctly in around 4200 s. But with 1 min of it, the program fails in 20 of 332 instances. Inside the 'out/' directory you can find the solutions for all problems.