

# Efficient External Memory Algorithms for Binary Decision Diagram Manipulation

---

**Steffan Christ Sølvesten**, Jaco van de Pol,  
Anna Blume Jakobsen, and Mathias Weller Berg Thomasen

November 26, 2021







**Figure 1:** The I/O model by Aggarwal and Vitter '87

For any realistic values of  $N$ ,  $M$ , and  $B$  we have that

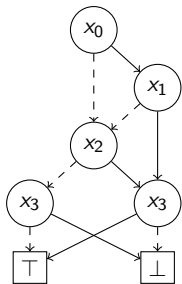
$$N/B < \text{sort}(N) \triangleq N/B \cdot \log_{M/B} N/B \ll N ,$$

**Theorem (Aggarwal and Vitter '87)**

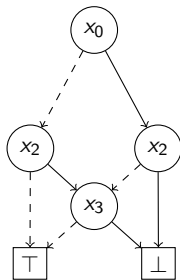
*$N$  elements can be sorted in  $\Theta(\text{sort}(N))$  I/Os.*

**Theorem (Arge '95)**

*$N$  elements can be inserted in and extracted from a Priority Queue in  $\Theta(\text{sort}(N))$  I/Os.*



**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

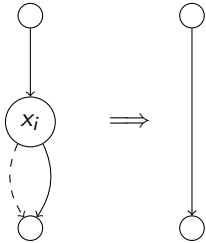


**(b)**  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

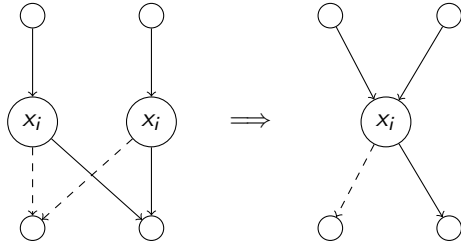
**Figure 2:** Examples of (Reduced Ordered) Binary Decision Diagrams.

### Theorem (Bryant '86)

*For a fixed variable order, if one exhaustively applies the two rules below, then one obtains the Reduced OBDD, which is a unique canonical form of the function.*

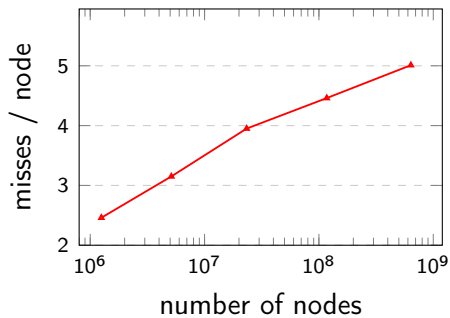
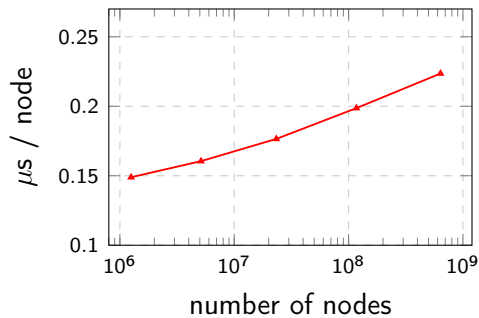


**(a)** Rule 1: Remove redundant nodes



**(b)** Rule 2: Merge duplicate nodes

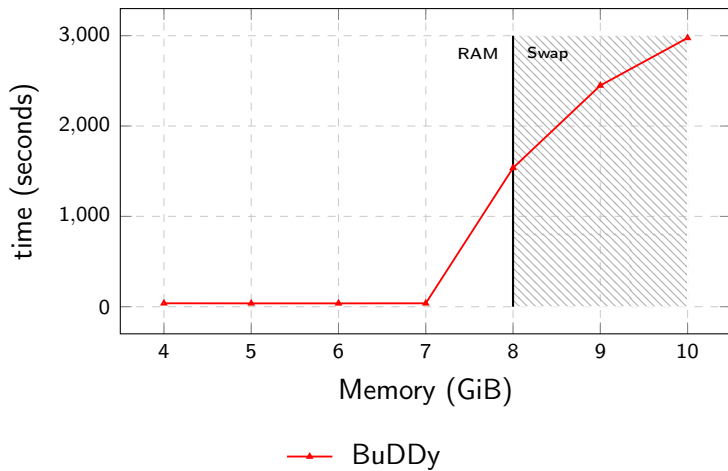




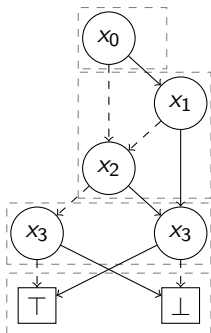
—▲— BuDDy

**Figure 4:** Cache behaviour for the  $N$ -Queens problem.





**Figure 5:** Running time for *Tic-Tac-Toe* with  $N = 21$ .

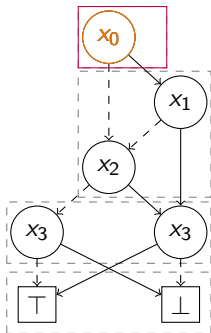


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
0	0

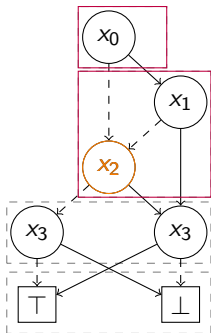


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
1	1

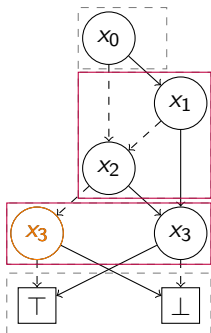


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
2	2

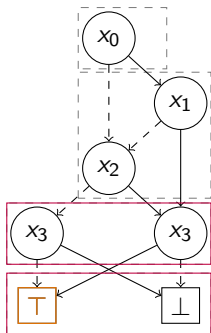


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
3	3

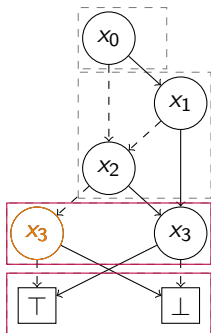


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
4	3

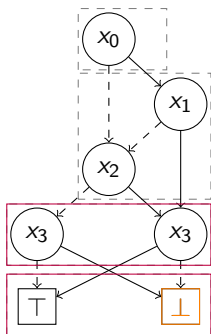


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
4	3



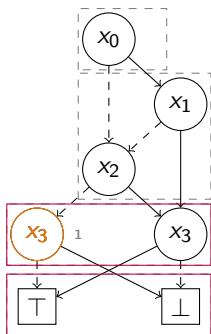
(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
4	3



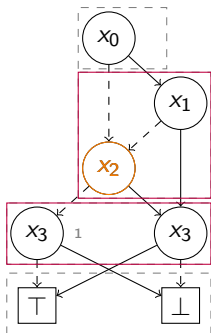


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
4	3

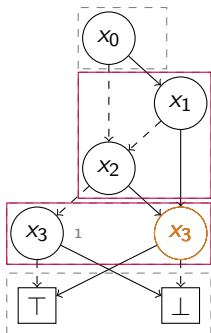


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
5	3

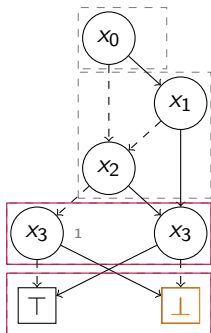


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
5	4

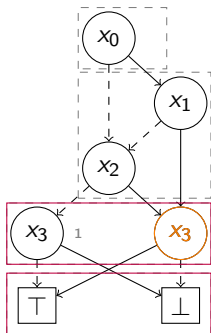


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
6	4

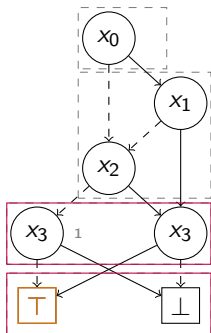


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
6	4

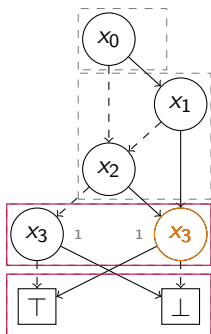


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
6	4

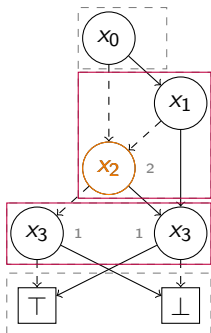


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
6	4



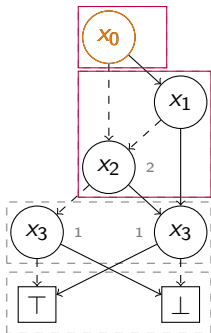
(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
7	4



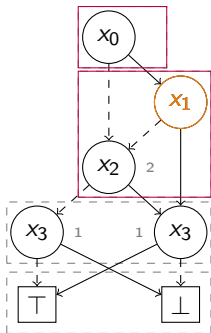


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
8	4

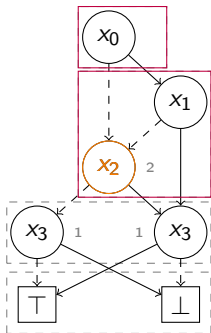


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
8	5

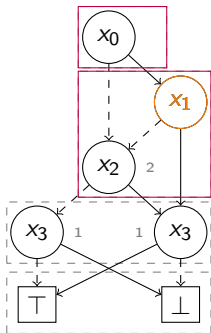


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
8	6

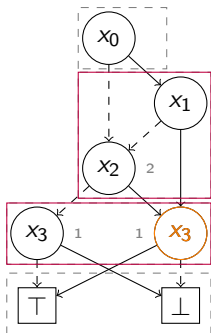


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
8	6

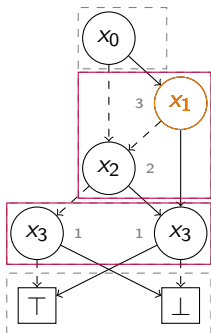


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
9	7

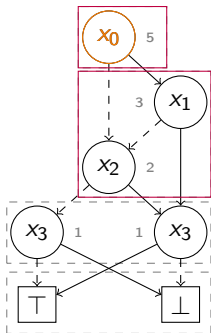


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
9	7

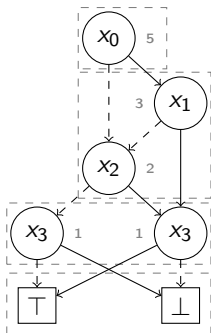


(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

$$M = 4, B = 2$$

node I/Os	cache lookups
10	7



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 6:** Blocks active in memory

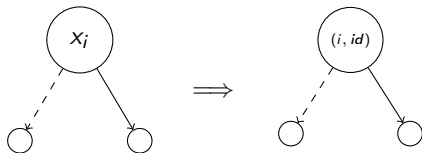
$$M = 4, B = 2$$

node I/Os	cache lookups
10	7



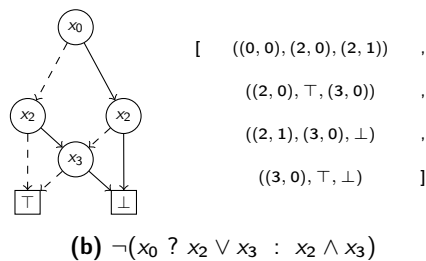
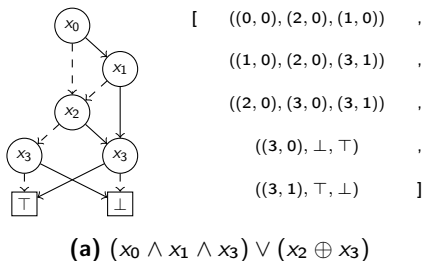


Let every node be uniquely identified by a tuple  $(label, id) : \mathbb{N} \times \mathbb{N}$ .



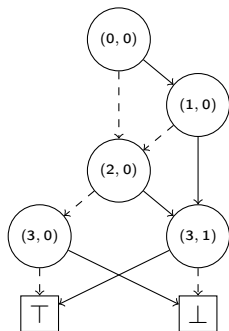
Nodes are ordered based on their *uid* as follows

$$(i_1, id_1) < (i_2, id_2) \equiv i_1 < i_2 \vee (i_1 = i_2 \wedge id_i < id_j)$$



**Figure 7:** Node-based representation of prior shown BDDs

# CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

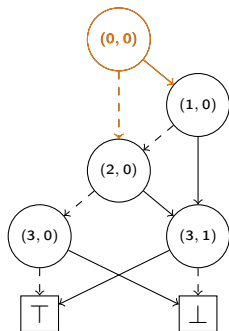
Priority Queue:  $Q_{count}$ :

[

]

**Figure 8:** In-order traversal of BDD

## CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

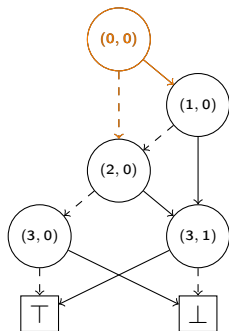
Priority Queue:  $Q_{count}$ :

[

]

**Figure 8:** In-order traversal of BDD

# CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

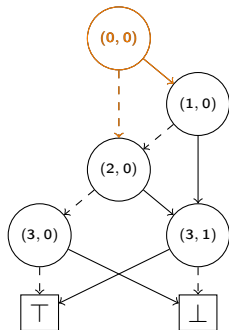
Priority Queue:  $Q_{count}$ :

[  $((0, 0) \xrightarrow{\top} (1, 0), 1)$  ,  
 $((0, 0) \xrightarrow{\perp} (2, 0), 1)$  ,

]

**Figure 8:** In-order traversal of BDD

## CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
(1, 0)	0	0

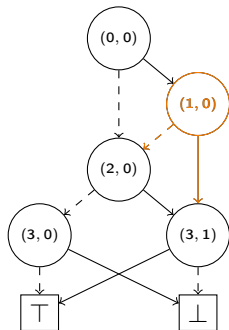
Priority Queue:  $Q_{count}$ :

[  $((0, 0) \xrightarrow{\top} (1, 0), 1)$  ,  
 $((0, 0) \xrightarrow{\perp} (2, 0), 1)$  ,

]

**Figure 8:** In-order traversal of BDD

## CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
(1, 0)	0	0

Priority Queue:  $Q_{count}$ :

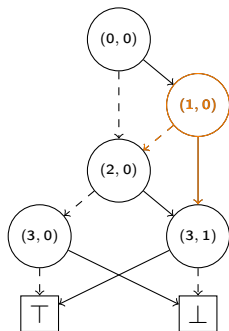
[  $((0, 0) \xrightarrow{\top} (1, 0), 1)$  ,  
 $((0, 0) \xrightarrow{\perp} (2, 0), 1)$  ,

]

**Figure 8:** In-order traversal of BDD



# CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
(1, 0)	1	0

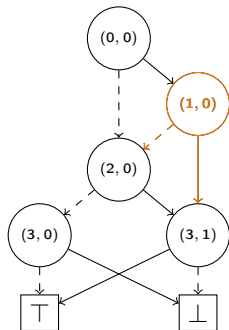
Priority Queue:  $Q_{count}$ :

[  
 $((0, 0) \xrightarrow{\perp} (2, 0), 1)$  ,

]

**Figure 8:** In-order traversal of BDD

## CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

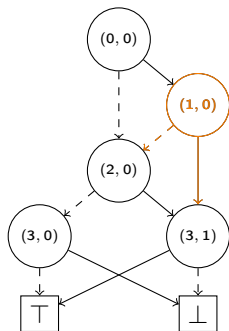
Seek	Sum	Result
(1, 0)	1	0

Priority Queue:  $Q_{count}$ :

[  
 $((0, 0) \xrightarrow{\perp} (2, 0), 1)$  ,  
 $((1, 0) \xrightarrow{\perp} (2, 0), 1)$  ,  
 $((1, 0) \xrightarrow{\top} (3, 1), 1)$  ,  
 ]

**Figure 8:** In-order traversal of BDD

## CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

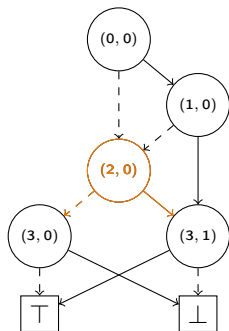
Seek	Sum	Result
(2, 0)	0	0

Priority Queue:  $Q_{count}$ :

[			
	$((0, 0) \xrightarrow{\perp} (2, 0), 1)$	,	
	$((1, 0) \xrightarrow{\perp} (2, 0), 1)$	,	
	$((1, 0) \xrightarrow{\top} (3, 1), 1)$	,	
			]

**Figure 8:** In-order traversal of BDD

## CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

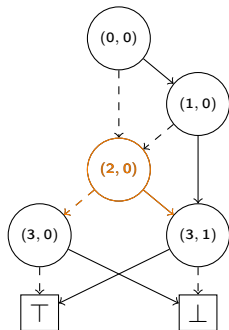
**Figure 8:** In-order traversal of BDD

Seek	Sum	Result
(2, 0)	0	0

Priority Queue:  $Q_{count}$ :

[			
	$((0, 0) \xrightarrow{\perp} (2, 0), 1)$	,	
	$((1, 0) \xrightarrow{\perp} (2, 0), 1)$	,	
	$((1, 0) \xrightarrow{\top} (3, 1), 1)$	,	
			]

## CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 8:** In-order traversal of BDD

Seek	Sum	Result
(2, 0)	1	0

Priority Queue:  $Q_{count}$ :

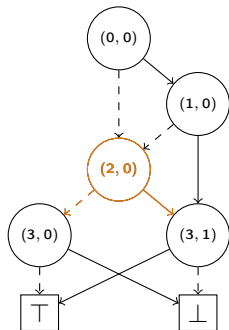
[

$((1, 0) \xrightarrow{\perp} (2, 0), 1)$  ,

$((1, 0) \xrightarrow{\top} (3, 1), 1)$  ,

]

## CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 8:** In-order traversal of BDD

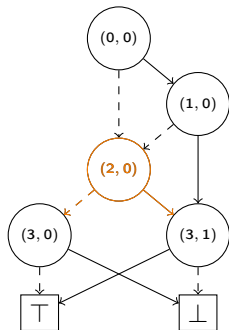
Seek	Sum	Result
(2, 0)	2	0

Priority Queue:  $Q_{count}$ :

[

$((1, 0) \xrightarrow{\top} (3, 1), 1)$  ,  
]

## CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
(2, 0)	2	0

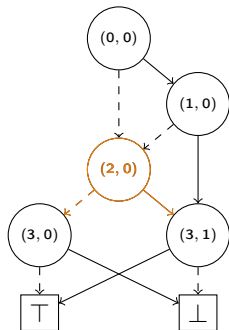
Priority Queue:  $Q_{count}$ :

[

$((2, 0) \xrightarrow{\perp} (3, 0), 2)$  ,  
 $((1, 0) \xrightarrow{\top} (3, 1), 1)$  ,  
 $((2, 0) \xrightarrow{\top} (3, 1), 2)$  ]

**Figure 8:** In-order traversal of BDD

## CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
(3, 0)	0	0

Priority Queue:  $Q_{count}$ :

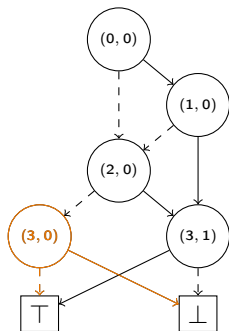
[

$((2, 0) \xrightarrow{\perp} (3, 0), 2)$  ,  
 $((1, 0) \xrightarrow{\top} (3, 1), 1)$  ,  
 $((2, 0) \xrightarrow{\top} (3, 1), 2)$  ]

**Figure 8:** In-order traversal of BDD



## CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
(3, 0)	0	0

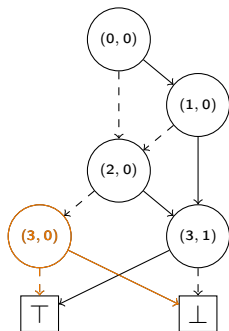
Priority Queue:  $Q_{count}$ :

[

$((2, 0) \xrightarrow{\perp} (3, 0), 2)$  ,  
 $((1, 0) \xrightarrow{\top} (3, 1), 1)$  ,  
 $((2, 0) \xrightarrow{\top} (3, 1), 2)$  ]

**Figure 8:** In-order traversal of BDD

## CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 8:** In-order traversal of BDD

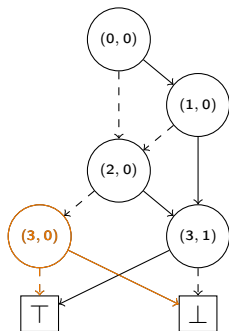
Seek	Sum	Result
(3, 0)	2	0

Priority Queue:  $Q_{count}$ :

[

$((1, 0) \xrightarrow{T} (3, 1), 1)$  ,  
 $((2, 0) \xrightarrow{T} (3, 1), 2)$  ]

## CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
(3, 0)	2	2

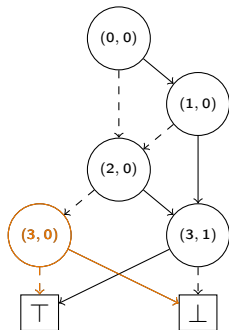
Priority Queue:  $Q_{count}$ :

[

$((1, 0) \xrightarrow{T} (3, 1), 1)$  ,  
 $((2, 0) \xrightarrow{T} (3, 1), 2)$  ]

**Figure 8:** In-order traversal of BDD

# CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
(3, 1)	0	2

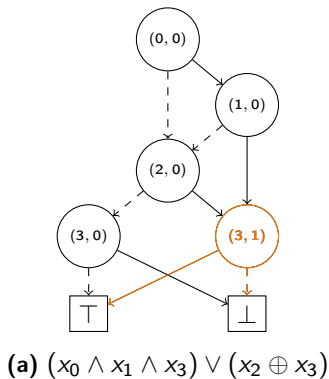
Priority Queue:  $Q_{count}$ :

[

$((1, 0) \xrightarrow{T} (3, 1), 1)$  ,  
 $((2, 0) \xrightarrow{T} (3, 1), 2)$  ]

**Figure 8:** In-order traversal of BDD

# CountPaths Example



**Figure 8:** In-order traversal of BDD

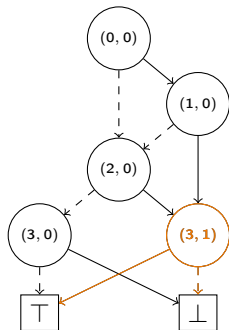
Seek	Sum	Result
(3, 1)	0	2

Priority Queue:  $Q_{count}$ :

[

$((1, 0) \xrightarrow{T} (3, 1), 1)$  ,  
 $((2, 0) \xrightarrow{T} (3, 1), 2)$  ]

## CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 8:** In-order traversal of BDD

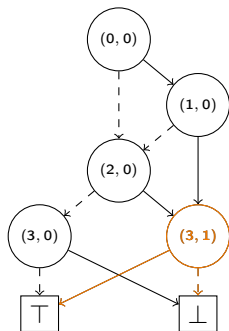
Seek	Sum	Result
(3, 1)	1	2

Priority Queue:  $Q_{count}$ :

[

$((2, 0) \xrightarrow{\top} (3, 1), \quad 2) \quad ]$

# CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 8:** In-order traversal of BDD

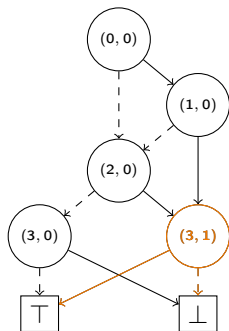
Seek	Sum	Result
(3, 1)	3	2

Priority Queue:  $Q_{count}$ :

[

]

# CountPaths Example



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

**Figure 8:** In-order traversal of BDD

Seek	Sum	Result
(3, 1)	3	5

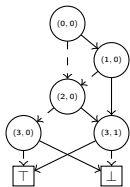
Priority Queue:  $Q_{count}$ :

[

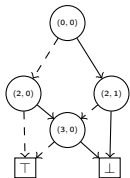
]



# Apply Example ( $\wedge$ )

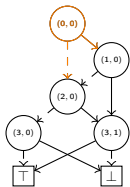


**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

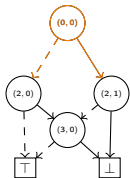


**(b)**  $\neg(x_0 \oplus x_2 \vee x_3 : x_2 \wedge x_3)$

# Apply Example ( $\wedge$ )

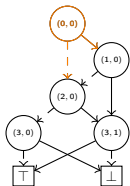


**(a)**  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

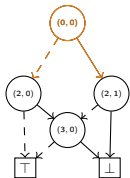


**(b)**  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

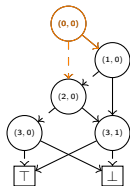
Priority Queue:  $Q_{app:1}$ :

[  $(0,0) \xrightarrow{\top} ((1,0), (2,1))$  ,  
 $(0,0) \xrightarrow{\perp} ((2,0), (2,0))$  ,

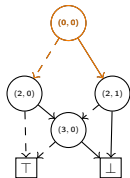


]

# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:

$\min((1, 0), (2, 1))$

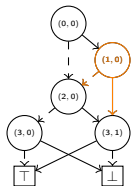
Priority Queue:  $Q_{app:1}$ :

[  $(0, 0) \xrightarrow{\top} ((1, 0), (2, 1))$  ,  
 $(0, 0) \xrightarrow{\perp} ((2, 0), (2, 0))$  ,

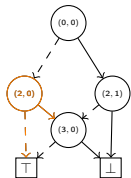


]

# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:

$\min((1, 0), (2, 1))$

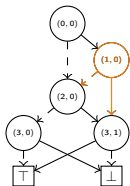
Priority Queue:  $Q_{app:1}$ :

[  $(0, 0) \xrightarrow{\top} ((1, 0), (2, 1))$  ,  
 $(0, 0) \xrightarrow{\perp} ((2, 0), (2, 0))$  ,

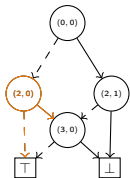


]

# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

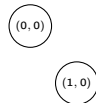
Seek:

$\min((1, 0), (2, 1))$

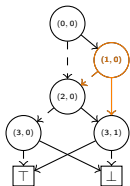
Priority Queue:  $Q_{app:1}$ :

[  $(0, 0) \xrightarrow{\top} ((1, 0), (2, 1))$  ,  
 $(0, 0) \xrightarrow{\perp} ((2, 0), (2, 0))$  ,  
 $(1, 0) \xrightarrow{\perp} ((2, 0), (2, 1))$  ,  
 $(1, 0) \xrightarrow{\top} ((3, 1), (2, 1))$  ,

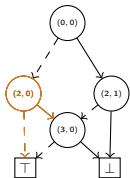
]



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:  
 $\min((1, 0), (2, 1))$

Priority Queue:  $Q_{app:1}$ :

[

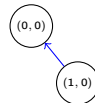
$(0, 0) \xrightarrow{\perp} ((2, 0), (2, 0))$  ,

$(1, 0) \xrightarrow{\perp} ((2, 0), (2, 1))$  ,

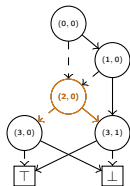
$(1, 0) \xrightarrow{\top} ((3, 1), (2, 1))$  ,

]

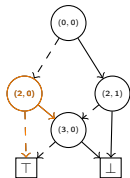
Output:  
 $(0, 0) \xrightarrow{\top} (1, 0)$



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:

$\min((2, 0), (2, 0))$

Priority Queue:  $Q_{app:1}$ :

[

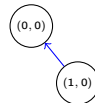
$(0, 0) \xrightarrow{\perp} ((2, 0), (2, 0))$  ,

$(1, 0) \xrightarrow{\perp} ((2, 0), (2, 1))$  ,

$(1, 0) \xrightarrow{\top} ((3, 1), (2, 1))$  ,

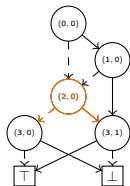
]

Output:

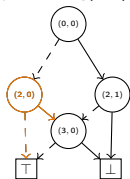




# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:

$\min((2, 0), (2, 0))$

Priority Queue:  $Q_{app:1}$ :

[

$(0, 0) \xrightarrow{\perp} ((2, 0), (2, 0))$  ,

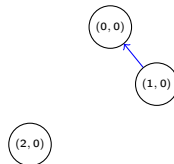
$(1, 0) \xrightarrow{\perp} ((2, 0), (2, 1))$  ,

$(1, 0) \xrightarrow{\top} ((3, 1), (2, 1))$  ,

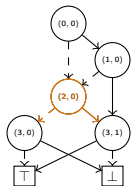
$(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,

$(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

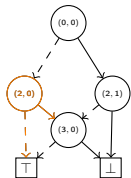
Output:



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:

$\min((2, 0), (2, 0))$

Priority Queue:  $Q_{app:1}$ :

[

$(1, 0) \xrightarrow{\perp} ((2, 0), (2, 1))$  ,

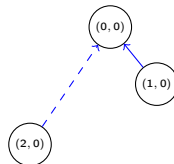
$(1, 0) \xrightarrow{\top} ((3, 1), (2, 1))$  ,

$(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,

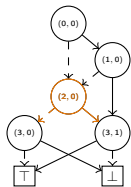
$(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

Output:

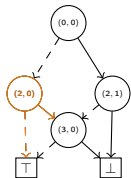
$(0, 0) \xrightarrow{\perp} (2, 0)$



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:

$\min((2, 0), (2, 1))$

Priority Queue:  $Q_{app:1}$ :

[

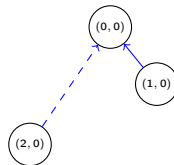
$(1, 0) \xrightarrow{\perp} ((2, 0), (2, 1))$  ,

$(1, 0) \xrightarrow{\top} ((3, 1), (2, 1))$  ,

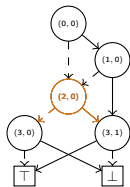
$(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,

$(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

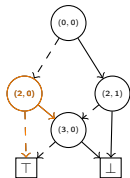
Output:



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:

$\min((2, 0), (2, 1))$

Priority Queue:  $Q_{app:1}$ :

[

$(1, 0) \xrightarrow{\top} ((3, 1), (2, 1))$  ,

$(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,

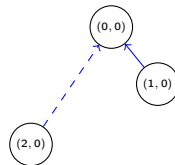
$(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

Priority Queue:  $Q_{app:2}$ :

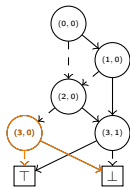
[  $(1, 0) \xrightarrow{\perp} ((2, 0), (2, 1))$   $((3, 0), (3, 1))$  ,

]

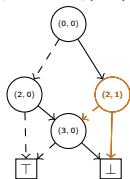
Output:



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:

$\max((2, 0), (2, 1))$

Priority Queue:  $Q_{app:1}$ :

[

$(1, 0) \xrightarrow{\top} ((3, 1), (2, 1))$  ,

$(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,

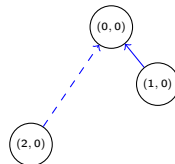
$(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

Priority Queue:  $Q_{app:2}$ :

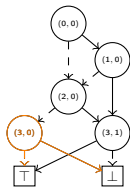
[  $(1, 0) \xrightarrow{\perp} ((2, 0), (2, 1))$   $((3, 0), (3, 1))$  ,

]

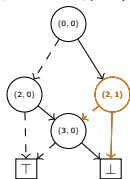
Output:



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:

$\max((2, 0), (2, 1))$

Priority Queue:  $Q_{app:1}$ :

[

$(1, 0) \xrightarrow{\top} ((3, 1), (2, 1))$  ,

$(2, 1) \xrightarrow{\perp} ((3, 0), (3, 0))$  ,

$(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,

$(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

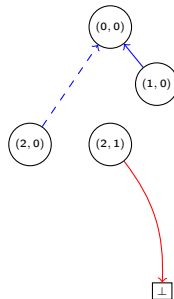
Priority Queue:  $Q_{app:2}$ :

[  $(1, 0) \xrightarrow{\perp} ((2, 0), (2, 1))$   $((3, 0), (3, 1))$  ,

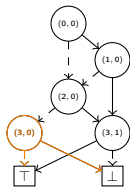
]

Output:

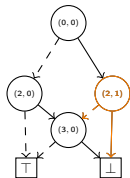
$(2, 1) \xrightarrow{\top} \perp$



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:  
 $\max((2, 0), (2, 1))$

Priority Queue:  $Q_{app:1}$ :

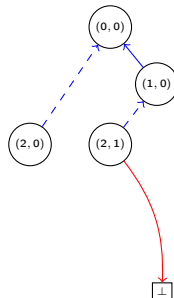
[

$(1, 0) \xrightarrow{\top} ((3, 1), (2, 1))$  ,  
 $(2, 1) \xrightarrow{\perp} ((3, 0), (3, 0))$  ,  
 $(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,

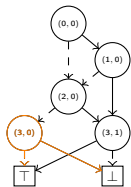
$(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

Priority Queue:  $Q_{app:2}$ :

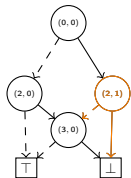
Output:  
 $(1, 0) \xrightarrow{\perp} (2, 1)$



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:  
 $\min((3, 1), (2, 1))$

Priority Queue:  $Q_{app:1}$ :

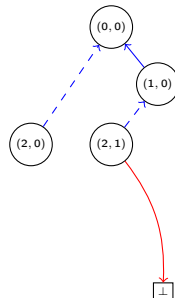
[

$(1, 0) \xrightarrow{\top} ((3, 1), (2, 1))$  ,  
 $(2, 1) \xrightarrow{\perp} ((3, 0), (3, 0))$  ,  
 $(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,

$(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

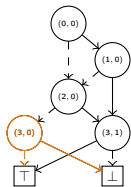
Priority Queue:  $Q_{app:2}$ :

Output:

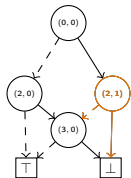




# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



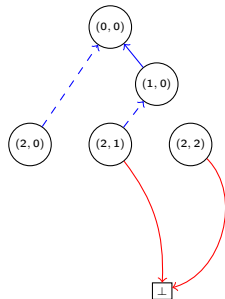
(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:  
 $\min((3, 1), (2, 1))$   
 Priority Queue:  $Q_{app:1}$ :  
 [

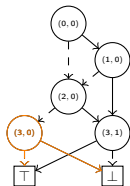
$(1, 0) \xrightarrow{\top} ((3, 1), (2, 1))$  ,  
 $(2, 1) \xrightarrow{\perp} ((3, 0), (3, 0))$  ,  
 $(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,  
 $(2, 2) \xrightarrow{\perp} ((3, 1), (3, 0))$  ,  
 $(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

Priority Queue:  $Q_{app:2}$ :

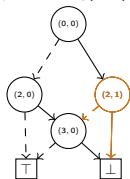
Output:  
 $(2, 2) \xrightarrow{\top} \perp$



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



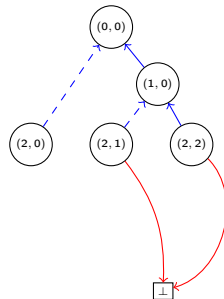
(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:  
 $\min((3, 1), (2, 1))$   
 Priority Queue:  $Q_{app:1}$ :  
 [

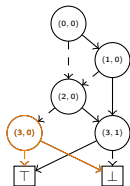
$(2, 1) \xrightarrow{\perp} ((3, 0), (3, 0))$  ,  
 $(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,  
 $(2, 2) \xrightarrow{\perp} ((3, 1), (3, 0))$  ,  
 $(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

Priority Queue:  $Q_{app:2}$ :

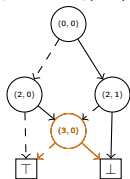
Output:  
 $(1, 0) \xrightarrow{\top} (2, 2)$



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:  
 $\min((3, 0), (3, 0))$

Priority Queue:  $Q_{app:1}$ :

[

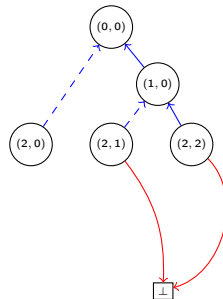
$(2, 1) \xrightarrow{\perp} ((3, 0), (3, 0))$  ,  
 $(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,  
 $(2, 2) \xrightarrow{\perp} ((3, 1), (3, 0))$  ,  
 $(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

Priority Queue:  $Q_{app:2}$ :

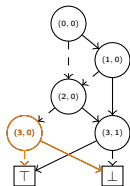
[

]

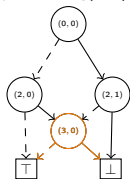
Output:



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:  
 $\min((3, 0), (3, 0))$

Priority Queue:  $Q_{app:1}$ :

[

$(2, 1) \xrightarrow{\perp} ((3, 0), (3, 0))$  ,  
 $(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,  
 $(2, 2) \xrightarrow{\perp} ((3, 1), (3, 0))$  ,  
 $(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

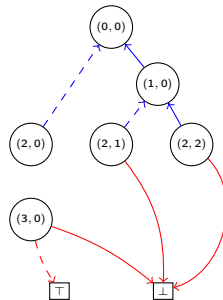
Priority Queue:  $Q_{app:2}$ :

[

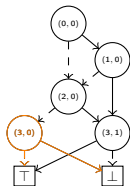
]

Output:

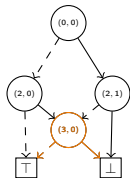
$(3, 0) \xrightarrow{\perp} \top, (3, 0) \xrightarrow{\top} \perp$



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:

$\min((3, 0), (3, 0))$

Priority Queue:  $Q_{app:1}$ :

[

$(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,

$(2, 2) \xrightarrow{\perp} ((3, 1), (3, 0))$  ,

$(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

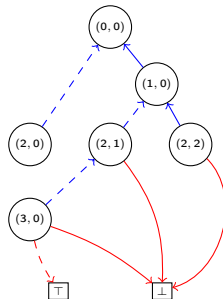
Priority Queue:  $Q_{app:2}$ :

[

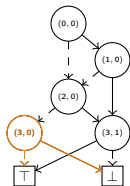
]

Output:

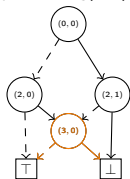
$(2, 1) \xrightarrow{\perp} (3, 0)$



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:  
 $\min((3, 1), (3, 0))$

Priority Queue:  $Q_{app:1}$ :

[

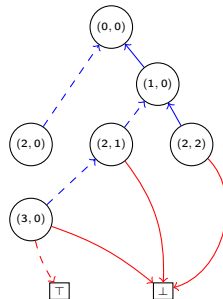
$(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,  
 $(2, 2) \xrightarrow{\perp} ((3, 1), (3, 0))$  ,  
 $(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

Priority Queue:  $Q_{app:2}$ :

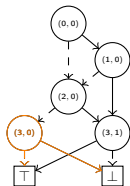
[

]

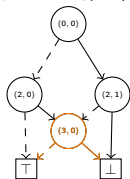
Output:



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:  
 $\min((3, 1), (3, 0))$

Priority Queue:  $Q_{app:1}$ :

[

$(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

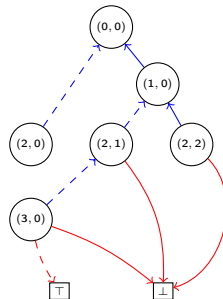
Priority Queue:  $Q_{app:2}$ :

[

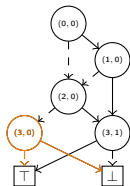
$(2, 0) \xrightarrow{\top} ((3, 1), (3, 0)) \quad (\top, \perp)$  ,

$(2, 2) \xrightarrow{\perp} ((3, 1), (3, 0)) \quad (\top, \perp)$  ]

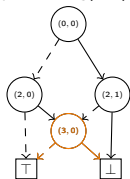
Output:



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:  
 $\min((3, 0), \top)$

Priority Queue:  $Q_{app:1}$ :

[

$(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

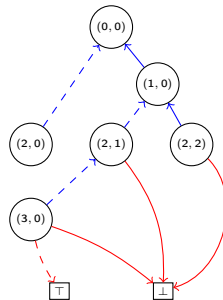
Priority Queue:  $Q_{app:2}$ :

[

$(2, 0) \xrightarrow{\top} ((3, 1), (3, 0)) \quad (\top, \perp)$  ,

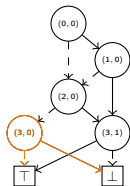
$(2, 2) \xrightarrow{\perp} ((3, 1), (3, 0)) \quad (\top, \perp)$  ]

Output:

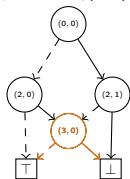




# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:  
 $\min((3, 0), \top)$

Priority Queue:  $Q_{app:1}$ :

[

$(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

Priority Queue:  $Q_{app:2}$ :

[

$(2, 0) \xrightarrow{\top} ((3, 1), (3, 0)) \quad (\top, \perp)$

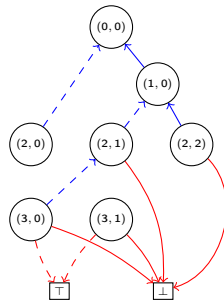
$(2, 2) \xrightarrow{\perp} ((3, 1), (3, 0)) \quad (\top, \perp)$

,

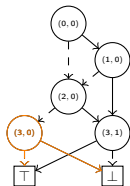
]

Output:

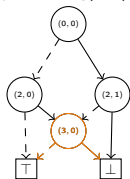
$(3, 1) \xrightarrow{\perp} \top, (3, 1) \xrightarrow{\top} \perp$



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:  
 $\min((3, 0), \top)$

Priority Queue:  $Q_{app:1}$ :

[

]

Priority Queue:  $Q_{app:2}$ :

[

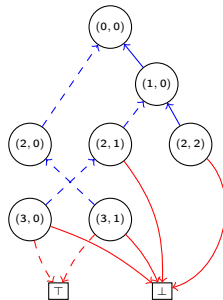
$(2, 0) \xrightarrow{\top} ((3, 1), (3, 0)) \quad (\top, \perp)$

$(2, 2) \xrightarrow{\perp} ((3, 1), (3, 0)) \quad (\top, \perp)$

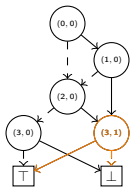
,

]

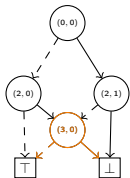
Output:  
 $(2, 0) \xRightarrow{\perp} (3, 1)$



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:  
 $\max((3, 1), (3, 0))$

Priority Queue:  $Q_{app:1}$ :

[

]

Priority Queue:  $Q_{app:2}$ :

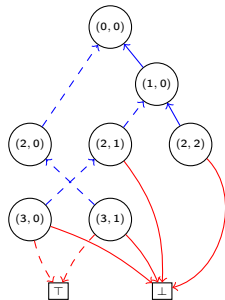
[

$(2, 0) \xrightarrow{\top} ((3, 1), (3, 0)) \quad (\top, \perp)$   
 $(2, 2) \xrightarrow{\perp} ((3, 1), (3, 0)) \quad (\top, \perp)$

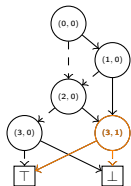
,

]

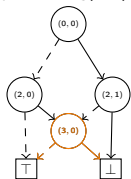
Output:



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

Seek:  
 $\max((3, 1), (3, 0))$

Priority Queue:  $Q_{app:1}$ :

[

]

Priority Queue:  $Q_{app:2}$ :

[

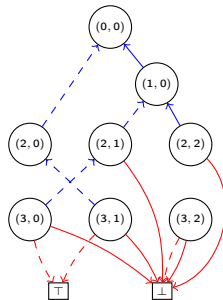
$(2, 0) \xrightarrow{\top} ((3, 1), (3, 0)) \quad (\top, \perp)$   
 $(2, 2) \xrightarrow{\perp} ((3, 1), (3, 0)) \quad (\top, \perp)$

,

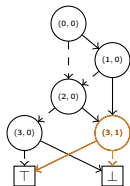
]

Output:

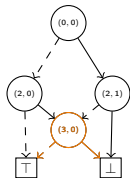
$(3, 2) \xrightarrow{\perp} \perp, (3, 2) \xrightarrow{\top} \perp$



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

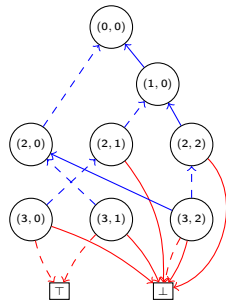


(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

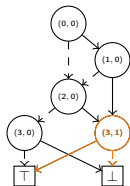
Seek:  
 $\max((3, 1), (3, 0))$   
 Priority Queue:  $Q_{app:1}$   
 [

Priority Queue:  $Q_{app:2}$   
 ]

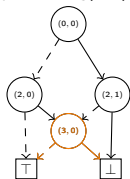
Output:  
 $(2, 0) \xrightarrow{T} (3, 2), (2, 2) \xrightarrow{\perp} (3, 2)$



# Apply Example ( $\wedge$ )



(a)  $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$



(b)  $\neg(x_0 ? x_2 \vee x_3 : x_2 \wedge x_3)$

[ Priority Queue:  $Q_{app:1}$ :

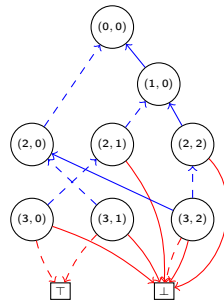
] ]

Priority Queue:  $Q_{app:2}$ :

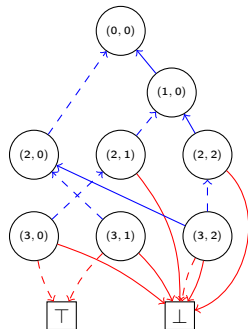
[

]

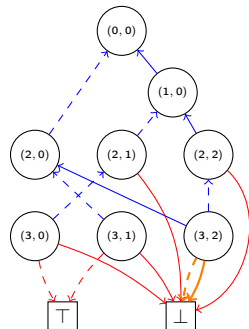
Output:



## Reduce Example



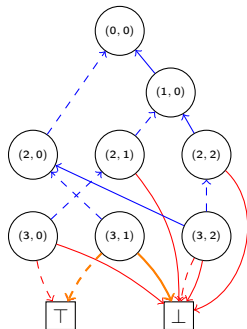
# Reduce Example



[                      Level: 3  
                     $[(3, 2) \mapsto \perp]$                       ]



# Reduce Example



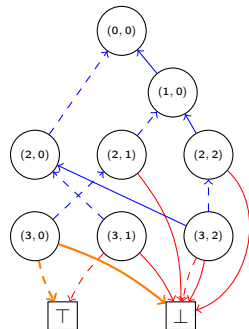
Level: 3

[  $[(3, 2) \mapsto \perp]$  ]

[  $((3, 1), \top, \perp)$  , ]

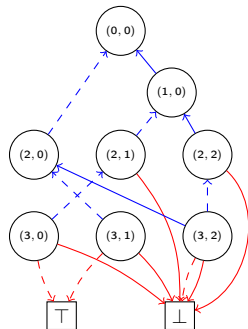
[ ]

# Reduce Example



Level: 3  
[  $[(3, 2) \mapsto \perp]$  ]  
[  $((3, 1), \top, \perp)$  ,  
  $((3, 0), \top, \perp)$  ]

# Reduce Example



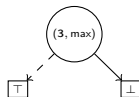
Level: 3

[  $[(3, 2) \mapsto \perp]$  ]

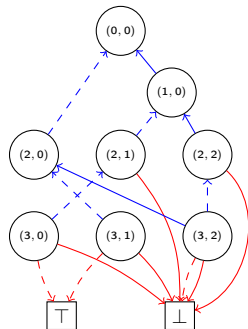
[  $[(3, 1) \mapsto (3, \max)]$  ,

$((3, 0), \top, \perp)$  ]

**Output:**  
 $((3, \max), \top, \perp)$



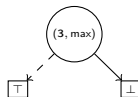
# Reduce Example



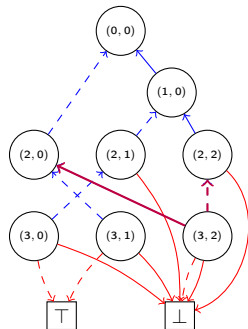
Level: 3

[	$[(3, 2) \mapsto \perp]$	]
[	$[(3, 1) \mapsto (3, \max)]$	,
	$[(3, 0) \mapsto (3, \max)]$	]

Output:



# Reduce Example



Priority Queue:  $Q_{red}$ :

[  $(2, 2) \xrightarrow{\perp} \perp$  ,

$(2, 0) \xrightarrow{T} \perp$  ,

]

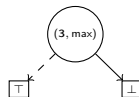
Level: 3

[ ]

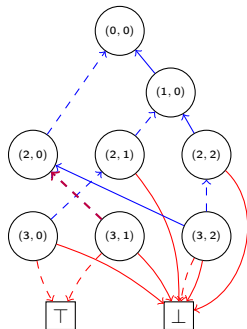
[  $[(3, 1) \mapsto (3, \max)]$  ,

$[(3, 0) \mapsto (3, \max)]$  ]

Output:



# Reduce Example



Priority Queue:  $Q_{red}$ :

[  $(2, 2) \xrightarrow{\perp} \perp$  ,

$(2, 0) \xrightarrow{\top} \perp$  ,

$(2, 0) \xrightarrow{\perp} (3, \max)$  ,

]

Level: 3

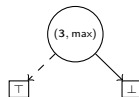
[ ]

[ ]

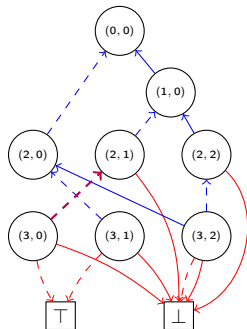
$[(3, 0) \mapsto (3, \max)]$

]

Output:



# Reduce Example



Priority Queue:  $Q_{red}$ :

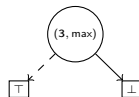
[  $(2, 2) \xrightarrow{\perp} \perp$  ,  
 $(2, 1) \xrightarrow{\perp} (3, \max)$  ,  
 $(2, 0) \xrightarrow{T} \perp$  ,  
 $(2, 0) \xrightarrow{\perp} (3, \max)$  ,

]

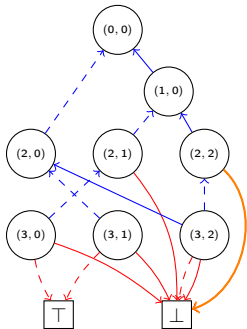
Level: 3

[  
 [  
 ,  
 ]

Output:



## Reduce Example



Priority Queue:  $Q_{red}$ :

[

$$(2, 1) \xrightarrow{\perp} (3, \max) \quad ,$$
$$(2, 0) \xrightarrow{T} \perp$$
$$(2, 0) \xrightarrow{\perp} (3, \max) \quad ,$$

1

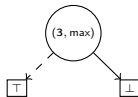
Level: 2

[

 $[(2, 2) \mapsto \perp]$ 

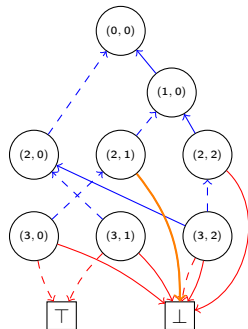
1

**Output:**





# Reduce Example



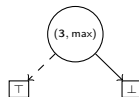
Priority Queue:  $Q_{red}$ :

[  
 $(2, 0) \xrightarrow{T} \perp$  ,  
 $(2, 0) \xrightarrow{\perp} (3, \max)$  ,  
 ]

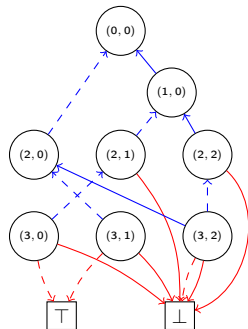
Level: 2

[  $[(2, 2) \mapsto \perp]$  ]  
 [  $((2, 1), (3, \max), \perp)$  ,  
 ]

Output:



# Reduce Example



Priority Queue:  $Q_{red}$ :

[

]

Level: 2

[

$[(2, 2) \mapsto \perp]$

]

[

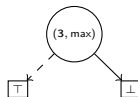
$((2, 1), (3, \max), \perp)$

,

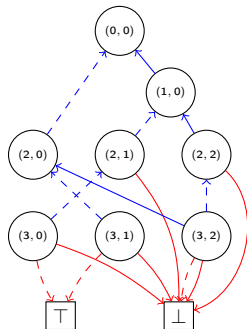
$((2, 0), (3, \max), \perp)$

]

Output:



# Reduce Example



Priority Queue:  $Q_{red}$ :

[

]

Level: 2

[

$[(2, 2) \mapsto \perp]$

]

[

$[(2, 1) \mapsto (2, \max)]$

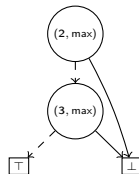
,

$((2, 0), (3, \max), \perp)$

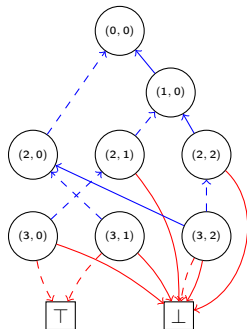
]

Output:

$((2, \max), (3, \max), \perp)$



# Reduce Example



Priority Queue:  $Q_{red}$ :

[

]

Level: 2

[

$[(2, 2) \mapsto \perp]$

]

[

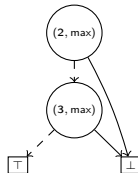
$[(2, 1) \mapsto (2, \max)]$

,

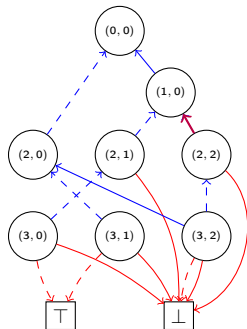
$[(2, 0) \mapsto (2, \max)]$

]

Output:



# Reduce Example



Priority Queue:  $Q_{red}$ :

[

$(1, 0) \xrightarrow{T} \perp$  ,

]

Level: 2

[

]

[

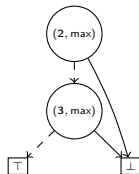
$[(2, 1) \mapsto (2, \max)]$

,

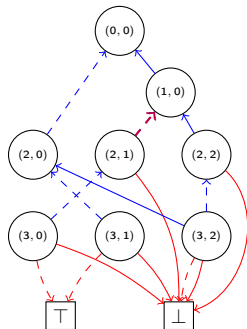
$[(2, 0) \mapsto (2, \max)]$

]

Output:



# Reduce Example



Priority Queue:  $Q_{red}$ :

[

$(1, 0) \xrightarrow{T} \perp$  ,

$(1, 0) \xrightarrow{\perp} (2, \max)$  ,

]

Level: 2

[

[

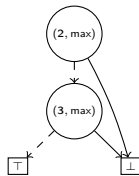
$[(2, 0) \mapsto (2, \max)]$

]

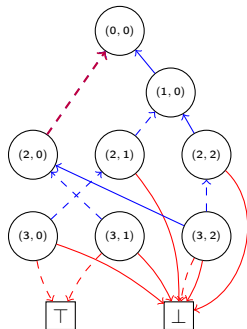
,

]

Output:



# Reduce Example



Priority Queue:  $Q_{red}$ :

[

$(1, 0) \xrightarrow{T} \perp$  ,

$(1, 0) \xrightarrow{\perp} (2, \max)$  ,

$(0, 0) \xrightarrow{\perp} (2, \max)$  ]

Level: 2

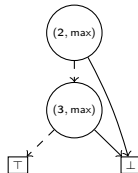
[

[

]

,  
]

Output:



# Reduce Example



Priority Queue:  $Q_{red}$ :

[

$(0, 0) \xrightarrow{\perp} (2, \max)$  ]

Level: 1

[

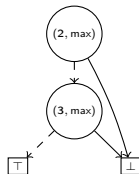
]

[

$((1, 0), (2, \max), \perp)$

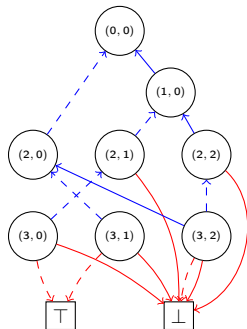
]

Output:





# Reduce Example



Priority Queue:  $Q_{red}$ :

[

$(0, 0) \xrightarrow{\perp} (2, \max)$  ]

Level: 1

[

]

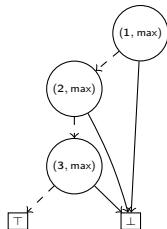
[

$[(1, 0) \mapsto (1, \max)]$

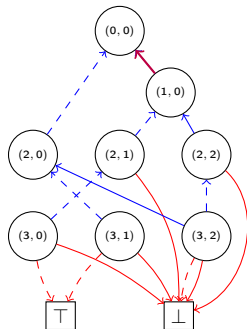
]

Output:

$((1, \max), (2, \max), \perp)$



# Reduce Example



Priority Queue:  $Q_{red}$ :

[

$(0, 0) \xrightarrow{T} (1, \max)$  ,

$(0, 0) \xrightarrow{\perp} (2, \max)$  ]

Level: 1

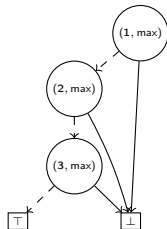
[

]

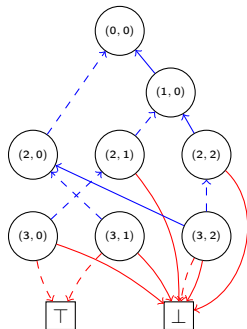
[

]

Output:



# Reduce Example



Priority Queue:  $Q_{red}$ :

[

]

Level: 0

[

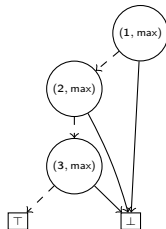
]

[

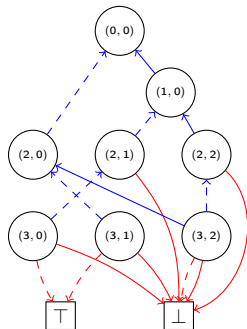
$((0, 0), (2, \max), (1, \max))$

]

Output:



# Reduce Example



Priority Queue:  $Q_{red}$ :

[

]

Level: 0

[

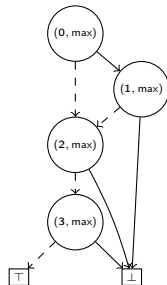
]

$[(0, 0) \mapsto (0, \max)]$

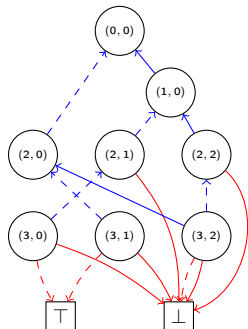
]

Output:

$((0, \max), (2, \max), (1, \max))$



# Reduce Example



Priority Queue:  $Q_{red}$ :

[

]

Level: 0

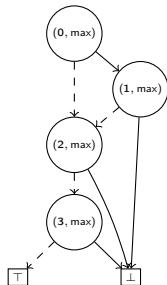
[

]

[

]

Output:





Levelized Priority Queue:  $Q_{app:1}$ :

Level: 1

$$[ \quad ]$$

Level: 2

[ , , ]

Level: 3

$$\left[ \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right]$$

Levelized Priority Queue:  $Q_{app:1}$ :

Level: 1

$$\left[ (0, 0) \xrightarrow{\top} ((1, 0), (2, 1)) \right]$$

Level: 2

$$\left[ (0, 0) \xrightarrow{\perp} ((2, 0), (2, 0)) \quad , \quad \quad \quad \right]$$

Level: 3

$$\left[ \quad , \quad , \quad , \quad \right]$$



Levelized Priority Queue:  $Q_{app:1}$ :

Level: 1

$$\left[ (0, 0) \xrightarrow{\top} ((1, 0), (2, 1)) \right]$$

Level: 2

$$\left[ (0, 0) \xrightarrow{\perp} ((2, 0), (2, 0)) \quad , \quad (1, 0) \xrightarrow{\perp} ((2, 0), (2, 1)) \quad , \quad (1, 0) \xrightarrow{\top} ((3, 1), (2, 1)) \right]$$

Level: 3

$$\left[ \quad , \quad , \quad , \quad \right]$$

Levelized Priority Queue:  $Q_{app:1}$ :

Level: 1

[ ]

Level: 2

[  $(0, 0) \xrightarrow{\perp} ((2, 0), (2, 0))$  ,  $(1, 0) \xrightarrow{\perp} ((2, 0), (2, 1))$  ,  $(1, 0) \xrightarrow{\top} ((3, 1), (2, 1))$  ]

Level: 3

[ , , , ]

Levelized Priority Queue:  $Q_{app:1}$ :

Level: 1

[ ]

Level: 2

[  $(0, 0) \xrightarrow{\perp} ((2, 0), (2, 0))$  ,  $(1, 0) \xrightarrow{\perp} ((2, 0), (2, 1))$  ,  $(1, 0) \xrightarrow{\top} ((3, 1), (2, 1))$  ]

Level: 3

[  $(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ,  $(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  , , ]

Levelized Priority Queue:  $Q_{app:1}$ :

Level: 1

[ ]

Level: 2

[ ,  $(1, 0) \xrightarrow{\perp} ((2, 0), (2, 1))$  ,  $(1, 0) \xrightarrow{\top} ((3, 1), (2, 1))$  ]

Level: 3

[  $(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ,  $(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  , ]

Levelized Priority Queue:  $Q_{app:1}$ :

Level: 1

[ ]

Level: 2

[ ,  $(1, 0) \xrightarrow{\perp} ((2, 0), (2, 1))$  ,  $(1, 0) \xrightarrow{\top} ((3, 1), (2, 1))$  ]

Level: 3

[  $(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ,  $(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,  $(2, 1) \xrightarrow{\perp} ((3, 0), (3, 0))$  , ]

Levelized Priority Queue:  $Q_{app:1}$ :

Level: 1

[ ]

Level: 2

[ , (1, 0)  $\xrightarrow{\top}$  ((3, 1), (2, 1)) ]

Level: 3

[ (2, 0)  $\xrightarrow{\perp}$  ((3, 0),  $\top$ ) , (2, 0)  $\xrightarrow{\top}$  ((3, 1), (3, 0)) , (2, 1)  $\xrightarrow{\perp}$  ((3, 0), (3, 0)) , ]

Levelized Priority Queue:  $Q_{app:1}$ :

Level: 1

[ ]

Level: 2

[ ,  $(1, 0) \xrightarrow{\top} ((3, 1), (2, 1))$  ]

Level: 3

[  $(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ,  $(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,  $(2, 1) \xrightarrow{\perp} ((3, 0), (3, 0))$  ,  $(2, 2) \xrightarrow{\perp} ((3, 1), (3, 0))$  ]

Levelized Priority Queue:  $Q_{app:1}$ :

Level: 1

[ ]

Level: 2

[ , ]

Level: 3

[  $(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ,  $(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,  $(2, 1) \xrightarrow{\perp} ((3, 0), (3, 0))$  ,  $(2, 2) \xrightarrow{\perp} ((3, 1), (3, 0))$  ]



Levelized Priority Queue:  $Q_{app:1}$ :

Level: 1

[ ]

Level: 2

[ , ]

Level: 3

[  $(2, 1) \xrightarrow{\perp} ((3, 0), (3, 0))$  ,  $(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,  $(2, 2) \xrightarrow{\perp} ((3, 1), (3, 0))$  ,  $(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

Levelized Priority Queue:  $Q_{app:1}$ :

Level: 1

[ ]

Level: 2

[ , ]

Level: 3

[ ,  $(2, 0) \xrightarrow{\top} ((3, 1), (3, 0))$  ,  $(2, 2) \xrightarrow{\perp} ((3, 1), (3, 0))$  ,  $(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

Levelized Priority Queue:  $Q_{app:1}$ :

Level: 1

[ ]

Level: 2

[ , ]

Level: 3

[ , (2, 2)  $\xrightarrow{\perp}$  ((3, 1), (3, 0)) , (2, 0)  $\xrightarrow{\perp}$  ((3, 0),  $\top$ ) ]

Levelized Priority Queue:  $Q_{app:1}$ :

Level: 1

[ ]

Level: 2

[ , , ]

Level: 3

[ , , ,  $(2, 0) \xrightarrow{\perp} ((3, 0), \top)$  ]

Levelized Priority Queue:  $Q_{app:1}$ :

Level: 1

$$[ \quad ]$$

Level: 2

[ , , ]

Level: 3

$$\left[ \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right]$$

## Memory layout and efficient sorting

The unique identifier of nodes and leafs can be represented in a single 64-bit integer.



(a) Unique identifier of a leaf  $v$



(b) Unique identifier of an internal node

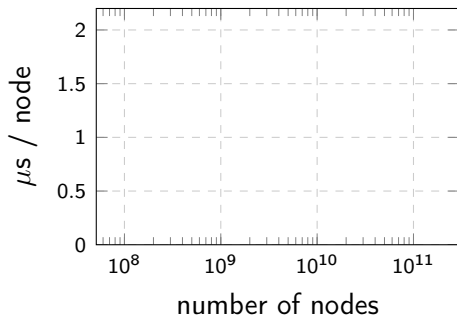
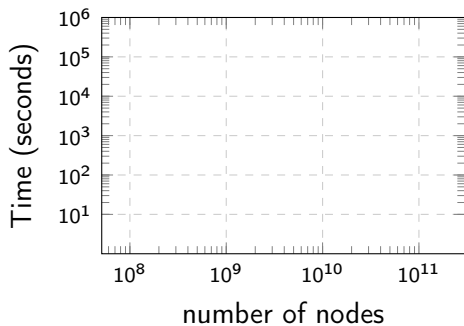
The  $f$  bit-flag is used to store the *is\_high* boolean inside of the source of an arc.



# Adiar

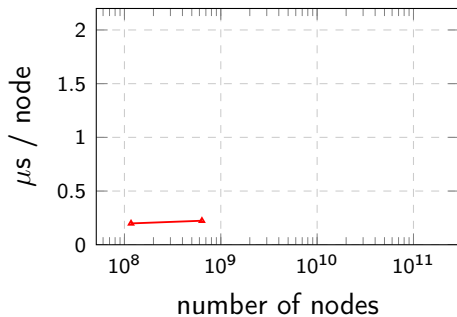
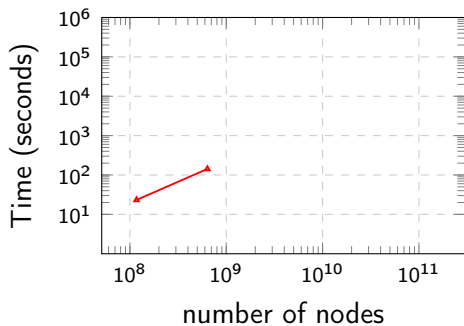
[github.com/ssoelvsten/adiar](https://github.com/ssoelvsten/adiar)





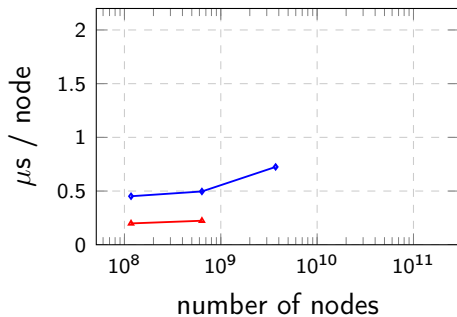
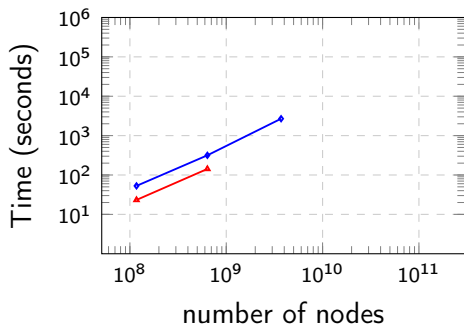
—▲— BuDDy —◆— CUDD —■— Sylvan —●— Adiar

**Figure 11:** Minimum running times for the  $N$ -Queens problem.



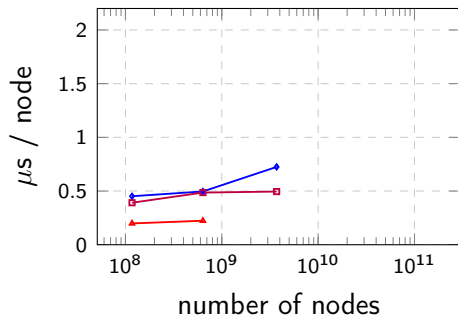
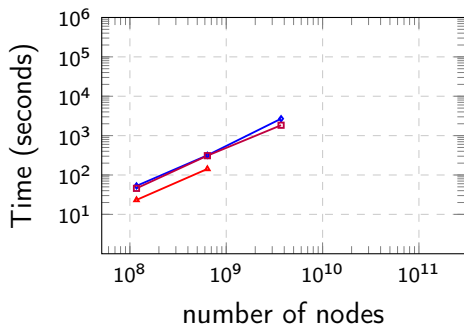
—▲— BuDDy —◆— CUDD —■— Sylvan —●— Adiar

**Figure 11:** Minimum running times for the  $N$ -Queens problem.



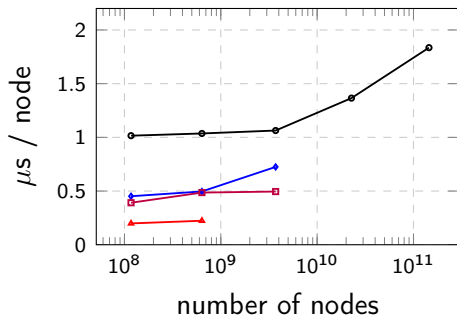
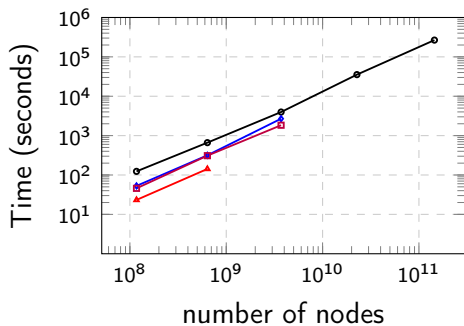
—▲— BuDDy —◆— CUDD —■— Sylvan —●— Adiar

**Figure 11:** Minimum running times for the  $N$ -Queens problem.



—▲— BuDDy —◆— CUDD —■— Sylvan —●— Adiar

**Figure 11:** Minimum running times for the  $N$ -Queens problem.



—▲— BuDDy —◆— CUDD —■— Sylvan —●— Adiar

**Figure 11:** Minimum running times for the  $N$ -Queens problem.



Algorithm		Depth-first	Time-forwarded
Reduce		$O(N)$	$O(\text{sort}(N))$
BDD Manipulation			
Apply	$f \odot g$	$O(N_f \cdot N_g)$	$O(\text{sort}(N_f \cdot N_g))$
If-Then-Else	$f ? g : h$	$O(N_f \cdot N_g \cdot N_h)$	$O(\text{sort}(N_f \cdot N_g \cdot N_h))$
Restrict	$f _{x_i=v}$	$O(N)$	$O(\text{sort}(N))$
Negation	$\neg f$	$O(1)$	$O(1)$
Quantification	$\exists/\forall v : f _{x_i=v}$	$O(N^2)$	$O(\text{sort}(N^2))$
Counting			
Count Paths	#paths in $f$ to $\top$	$O(N)$	$O(\text{sort}(N))$
Count SAT	$\#x : f(x)$	$O(N)$	$O(\text{sort}(N))$
Other			
Equality	$f \equiv g$	$O(1)$	$O(\text{sort}(N))$
Evaluate	$f(x)$	$O(L)$	$O(N/B)$
Min/Max SAT	$\min / \max\{x \mid f(x)\}$	$O(L)$	$O(N/B)$

**Table 1:** I/O-complexity of depth-first algorithms compared to our time-forwarded.

