Random Access on Narrow Decision Diagrams in External Memory

Steffan Christ Sølvsten, Casper Moldrup Rysgaard, and Jaco van de Pol

SPIN 2024

AARHUS
UNIVERSITY

Adiar

I/O-efficient Decision Diagrams

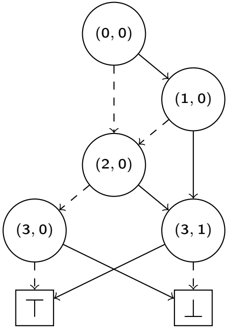
github.com/ssoelvsten/adiar

Features

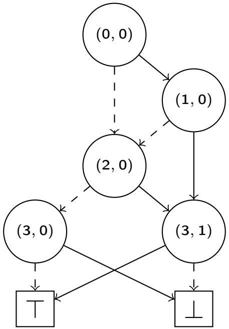
Optimisations

Features

Optimisations



(a) $(x_0 \land x_1 \land x_3) \lor (x_2 \oplus x_3)$

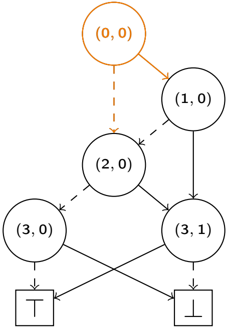


(a) $(x_0 \land x_1 \land x_3) \lor (x_2 \oplus x_3)$

Priority Queue: Q_{count}:

[

]

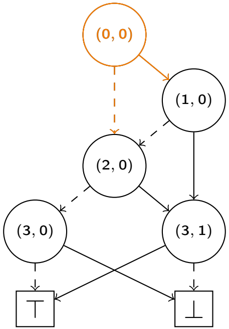


(a) $(x_0 \land x_1 \land x_3) \lor (x_2 \oplus x_3)$

Priority Queue: Q_{count}:

[

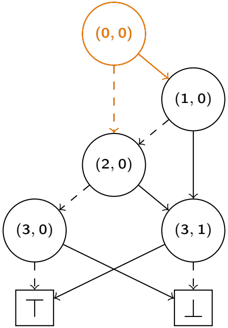
]



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

Priority Queue: Q_{count}:

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

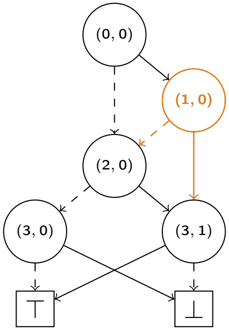


(a) $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
$(1,0)$	0	0

Priority Queue: Q_{count}:

$$\begin{array}{llll} [& ((0,0) \xrightarrow{\top} (1,0), & 1) & , \\ & ((0,0) \xrightarrow{\perp} (2,0), & 1) & , \\ & & &] \end{array}$$



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

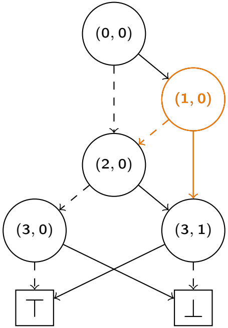
Seek	Sum	Result
$(1, 0)$	0	0

Priority Queue: Q_{count}:

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

$((0,0) \xrightarrow{\bot} (2,0), \quad 1) \quad ,$

$]$



(a) $(x_0 \land x_1 \land x_3) \lor (x_2 \oplus x_3)$

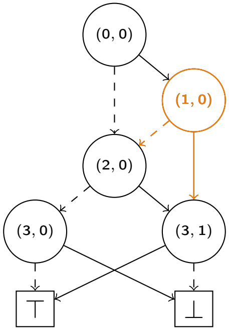
Seek	Sum	Result
$(1, 0)$	1	0

Priority Queue: Q_{count}:

[

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

]

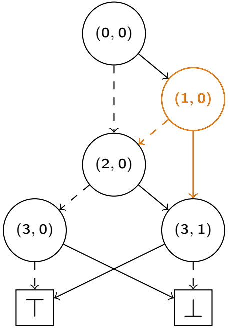


(a) $(x_0 \land x_1 \land x_3) \lor (x_2 \oplus x_3)$

Seek	Sum	Result
$(1, 0)$	1	0

Priority Queue: Q_{count}:

$$
\begin{array}{l}
[\\
\quad ((0,0) \xrightarrow{\bot} (2,0), \quad 1) \quad , \\
\quad ((1,0) \xrightarrow{\bot} (2,0), \quad 1) \quad , \\
\\
\quad ((1,0) \xrightarrow{\top} (3,1), \quad 1) \quad , \\
\quad]
\end{array}
$$



(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}:

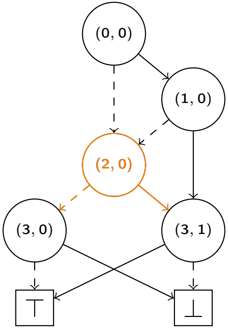
[

$((0,0) \xrightarrow{\bot} (2,0),\quad 1)$,

$((1,0) \xrightarrow{\bot} (2,0),\quad 1)$,

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

]



(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}:

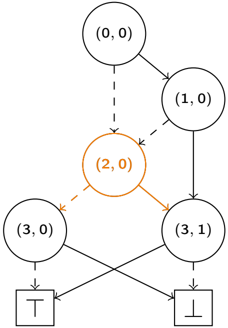
$[$

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

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

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

$]$



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

Seek	Sum	Result
$(2, 0)$	1	0

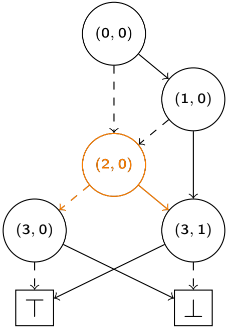
Priority Queue: Q_{count}:

[

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

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

]



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

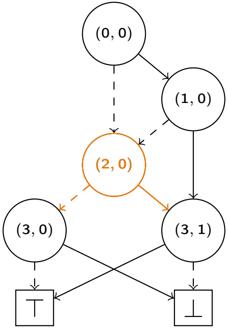
Seek	Sum	Result
$(2,0)$	2	0

Priority Queue: Q_{count}:

[

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

]

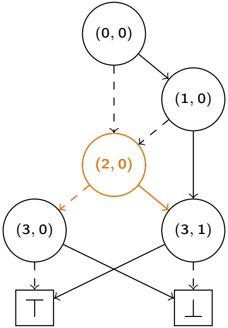


(a) $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
$(2, 0)$	2	0

Priority Queue: Q_{count}:

$$
\begin{aligned}
[\; & ((2,0) \xrightarrow{\perp} (3,0), \quad 2) \quad , \\
& ((1,0) \xrightarrow{\top} (3,1), \quad 1) \quad , \\
& ((2,0) \xrightarrow{\top} (3,1), \quad 2) \quad]
\end{aligned}
$$

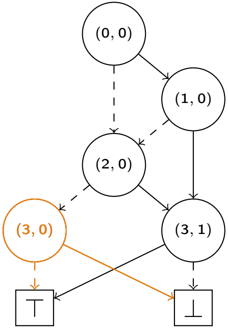


(a) $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
(3, 0)	0	0

Priority Queue: Q_{count}:

$$
\begin{array}{lllll}
[& & & & \\
 & ((2,0) \xrightarrow{\bot} (3,0), & 2) & & , \\
 & ((1,0) \xrightarrow{\top} (3,1), & 1) & & , \\
 & ((2,0) \xrightarrow{\top} (3,1), & 2) & &]
\end{array}
$$



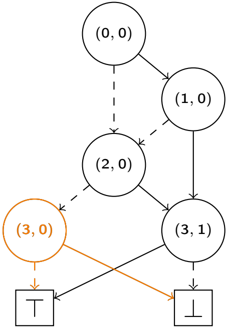
(a) $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
(3, 0)	0	0

Priority Queue: Q_{count}:

[

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

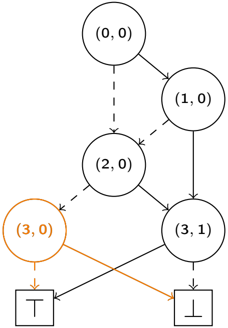


(a) $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
$(3, 0)$	2	0

Priority Queue: Q_{count}:

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



(a) $(x_0 \land x_1 \land x_3) \lor (x_2 \oplus x_3)$

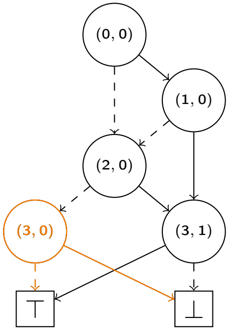
Seek	Sum	Result
(3, 0)	2	2

Priority Queue: Q_{count}:

[

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

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



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

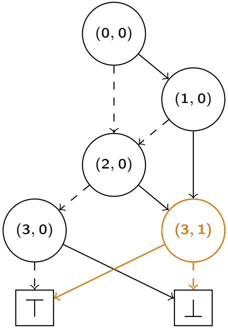
Seek	Sum	Result
$(3, 1)$	0	2

Priority Queue: Q_{count}:

[

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

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



(a) $(x_0 \land x_1 \land x_3) \lor (x_2 \oplus x_3)$

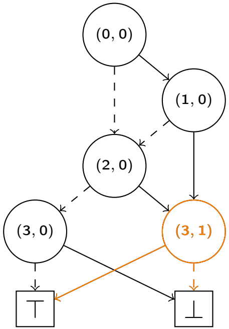
Seek	Sum	Result
(3, 1)	0	2

Priority Queue: Q_{count}:

[

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

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



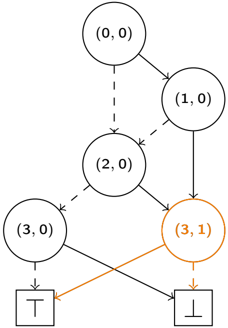
(a) $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
$(3, 1)$	1	2

Priority Queue: Q_{count}:

[

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



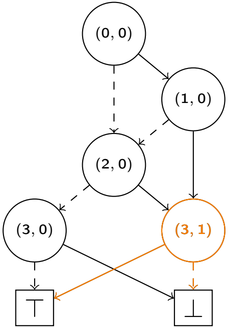
(a) $(x_0 \land x_1 \land x_3) \lor (x_2 \oplus x_3)$

Seek	Sum	Result
$(3, 1)$	3	2

Priority Queue: Q_{count}:

[

]



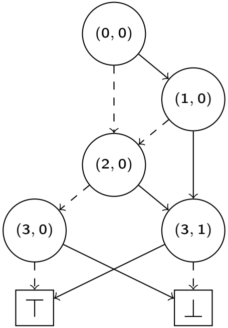
(a) $(x_0 \wedge x_1 \wedge x_3) \vee (x_2 \oplus x_3)$

Seek	Sum	Result
$(3, 1)$	3	5

Priority Queue: Q_{count}:

[

]



(a) $(x_0 \land x_1 \land x_3) \lor (x_2 \oplus x_3)$

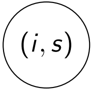
Result

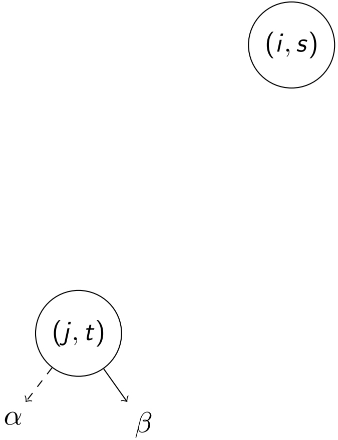
5

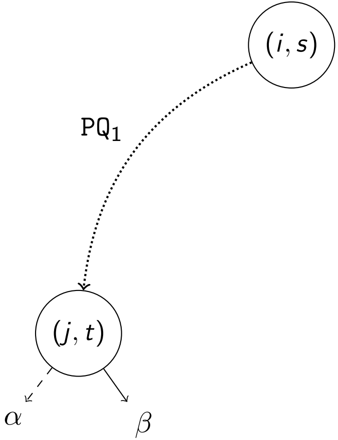
Priority Queue: Q_{count}:

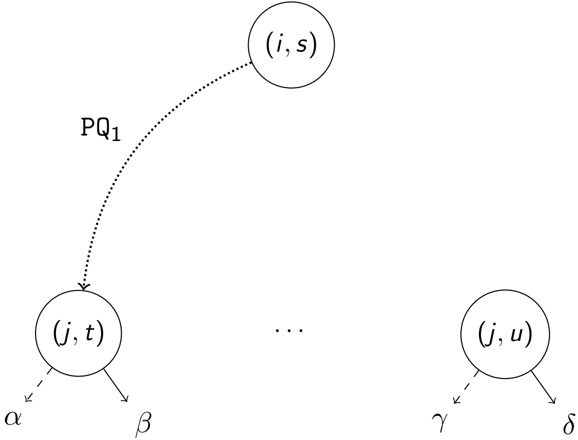
[

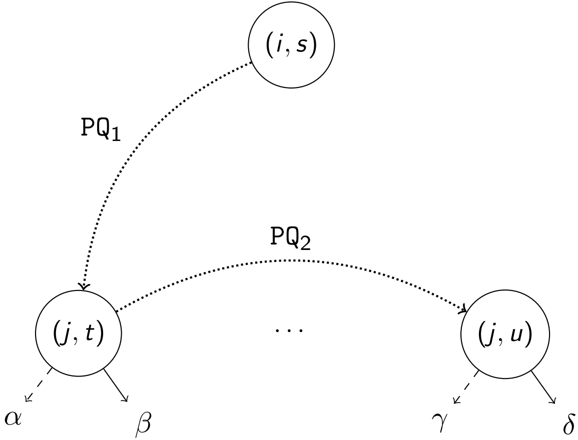
]

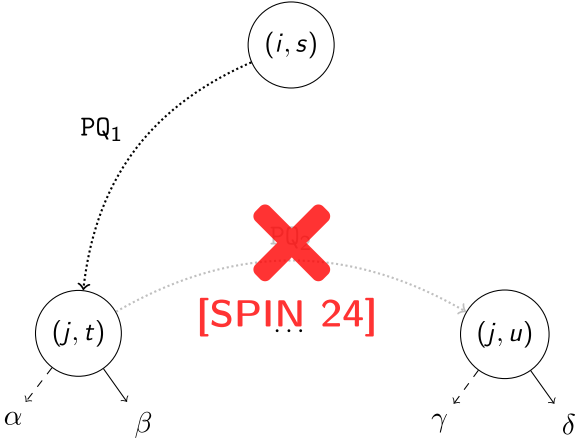

(i, s)


(i, s)
(j, t)
α
β

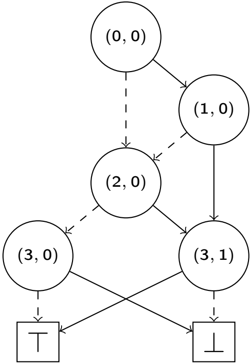

(i, s)
$\mathtt{PQ}_1$
(j, t)
α
β

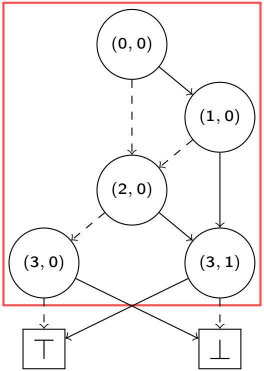

(i, s)
$\texttt{PQ}_1$
(j, t)
$\cdots$
(j, u)
α
β
γ
δ

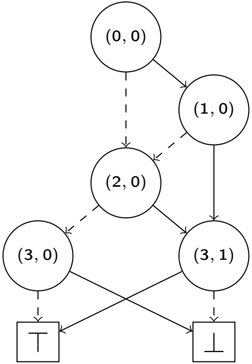

(i, s)
$\texttt{PQ}_1$
$\texttt{PQ}_2$
(j, t)
$\cdots$
(j, u)
α
β
γ
δ

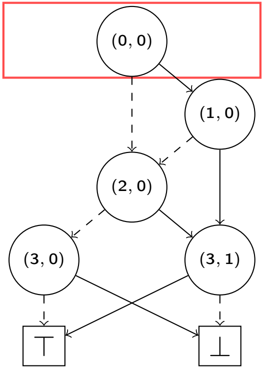


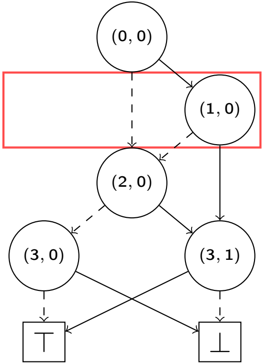
[SPIN 24]

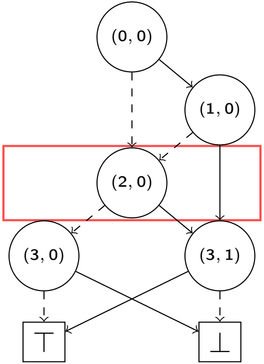

(0, 0)
(1, 0)
(2, 0)
(3, 0)
(3, 1)
⊤
⊥

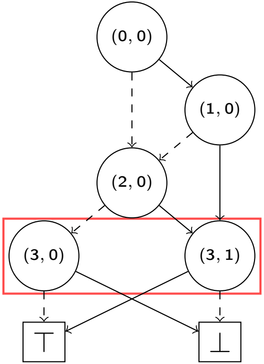

(0, 0)
(1, 0)
(2, 0)
(3, 0)
(3, 1)
⊤
⊥

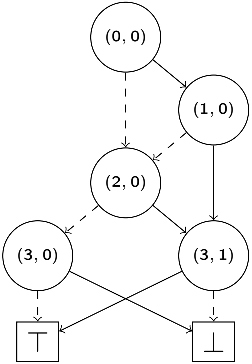

(0, 0)
(1, 0)
(2, 0)
(3, 0)
(3, 1)
⊤
⊥

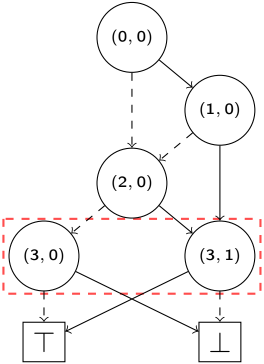

(0, 0)
(1, 0)
(2, 0)
(3, 0)
(3, 1)
⊤
⊥


(0, 0)
(1, 0)
(2, 0)
(3, 0)
(3, 1)
⊤
⊥

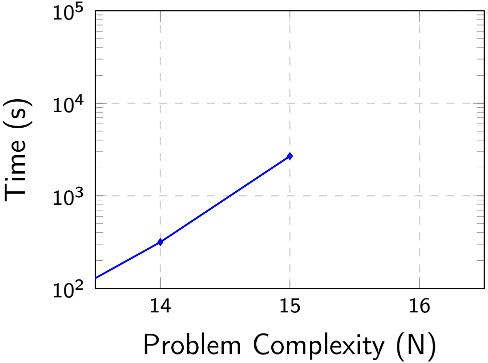

(0, 0)
(1, 0)
(2, 0)
(3, 0)
(3, 1)
⊤
⊥


(0, 0)
(1, 0)
(2, 0)
(3, 0)
(3, 1)
⊤
⊥


(0, 0)
(1, 0)
(2, 0)
(3, 0)
(3, 1)
⊤
⊥

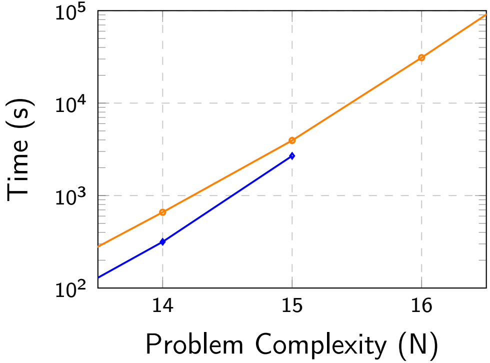


$$\frac{\text{Width}}{2}$$



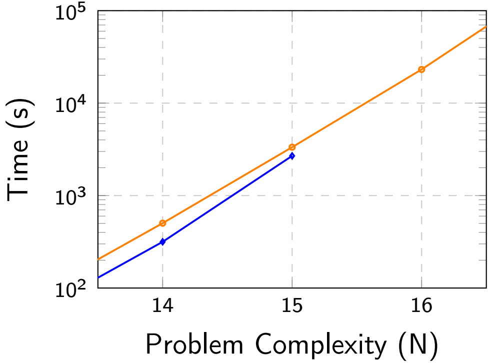
				⏱ $N = 15$
◇	CUDD	v3.0	:	44.8 min

Queens | 300 GiB of RAM



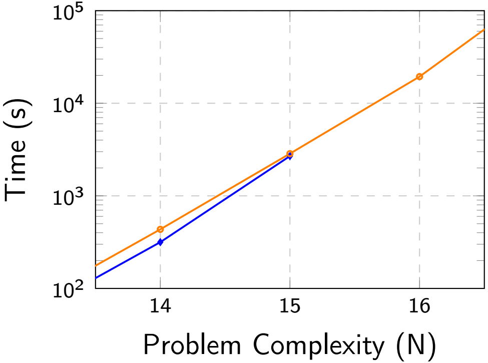
				⏱ $N = 15$
◇	CUDD	v3.0	:	44.8 min
○	Adiar	v1.0	:	66.7 min

Queens | 300 GiB of RAM



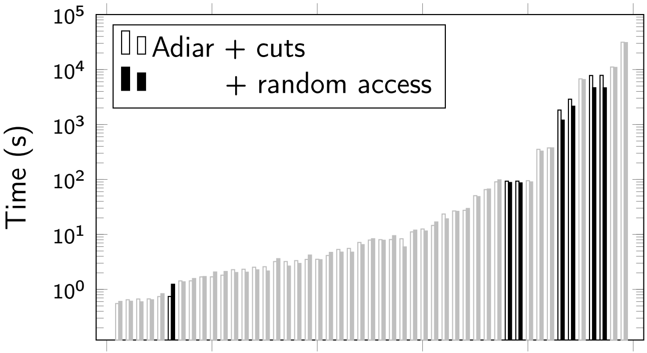
				⏱ $N = 15$
◇	CUDD	v3.0	:	44.8 min
○	Adiar	v1.0	:	66.7 min
	+ cuts		:	56.8 min

Queens | 300 GiB of RAM



				⏱ $N = 15$
◇	CUDD	v3.0	:	44.8 min
○	Adiar	v1.0	:	66.7 min
	+ cuts		:	56.8 min
	+ random access		:	47.2 min

Queens | 300 GiB of RAM



⌃

Speed Ups

6 (+26.9%)

⌄

Slowdowns

1 (-66.8%)

EPFL Circuit Verification | 300 GiB of RAM

Steffan Christ Sølvsten

✉ `soelvsten@cs.au.dk`

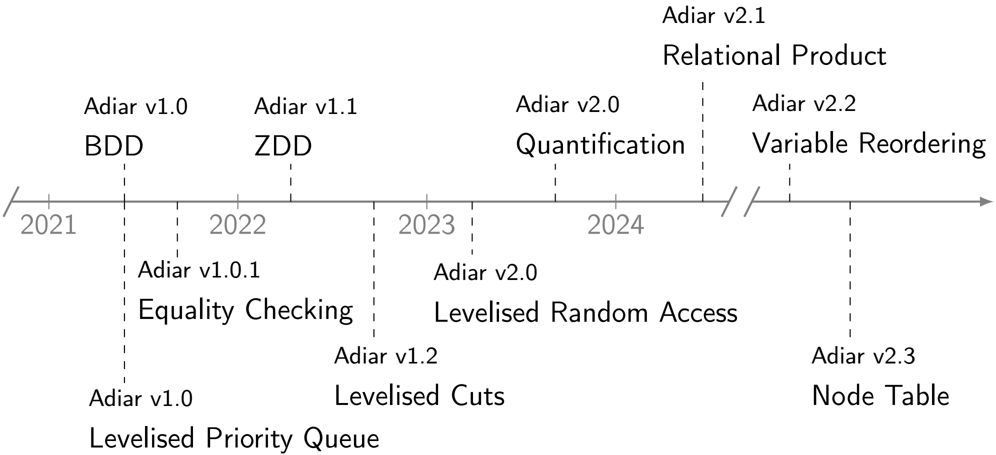
🌐 ssoelvsten.github.io

Adiar

</> github.com/ssoelvsten/adiar

📖 ssoelvsten.github.io/adiar

AARHUS
UNIVERSITY



Adiar v1.0
BDD

Adiar v1.0
Levelised Priority Queue

Adiar v1.0.1
Equality Checking

Adiar v1.1
ZDD

Adiar v1.2
Levelised Cuts

Adiar v2.0
Levelised Random Access

Adiar v2.0
Quantification

Adiar v2.1
Relational Product

Adiar v2.2
Variable Reordering

Adiar v2.3
Node Table