Knowledge Representation for Quantum Computing

Santiago $\longleftrightarrow$ 2026 • Lecture 3 of 3

Alfons Laarman

Leiden University
Santiago 2026

Universiteit Leiden
The Netherlands

Santiago 2026 — a three-lecture series

1. Demystifying Quantum Computing — *quantum computing 101*
2. Quantum Circuit Analysis with #SAT — *model counting*
3. **Knowledge Representation for Quantum Computing** — *knowledge representation*

Knowledge Compilation Map

A knowledge compilation map [1]

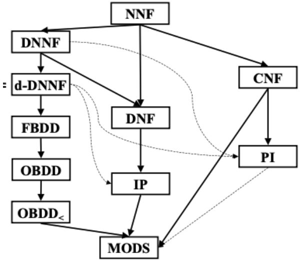
- Normal forms for representing Boolean functions: $f\colon \{0,1\}^n \to \{0,1\}$
- Tradeoff: succinctness vs tractability

[1] A. Darwiche and P. Marquis, "A knowledge compilation map," *Journal of Artificial Intelligence Research*, vol. 17, 2002.

Knowledge Compilation Map

A knowledge compilation map [1]

- Normal forms for representing Boolean functions: $f\colon \{0,1\}^n \to \{0,1\}$
- Tradeoff: succinctness vs tractability



Succinctness relations
(exponential separations)

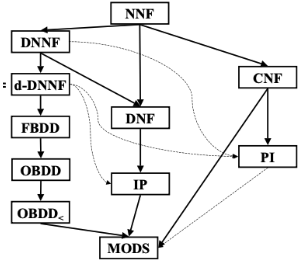
L	CD	FO	SFO	$\wedge$C	$\wedge$BC	$\vee$C	$\vee$BC	$\neg$C
NNF	√	○	√	√	√	√	√	√
DNNF	√	√	√	○	○	√	√	○
d-NNF	√	○	√	√	√	√	√	√
s-NNF	√	○	√	√	√	√	√	√
f-NNF	√	○	√	•	•	•	•	√
d-DNNF	√	○	○	○	○	○	○	?
sd-DNNF	√	○	○	○	○	○	○	?
BDD	√	○	√	√	√	√	√	√
FBDD	√	•	○	•	○	•	○	√
OBDD	√	•	√	•	○	•	○	√
$\text{OBDD}_<$	√	•	√	•	√	•	√	√
DNF	√	√	√	•	√	√	√	•
CNF	√	○	√	√	√	•	√	•
PI	√	√	√	•	•	•	√	•
IP	√	•	•	•	√	•	•	•
MODS	√	√	√	•	√	•	•	•

operation tractability

[1] A. Darwiche and P. Marquis, "A knowledge compilation map," *Journal of Artificial Intelligence Research*, vol. 17, 2002.

A knowledge compilation map [1]

- Normal forms for representing Boolean functions: $f\colon \{0,1\}^n \rightarrow \{0,1\}$
- Tradeoff: succinctness vs tractability
- New domain: A quantum state is a pseudo-Boolean function $f\colon \{0,1\}^n \rightarrow \mathbb{C}$

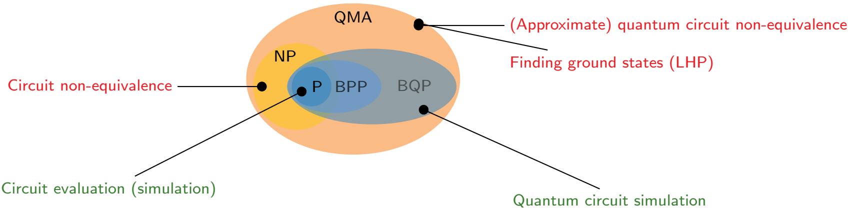


Succinctness relations
(exponential separations)

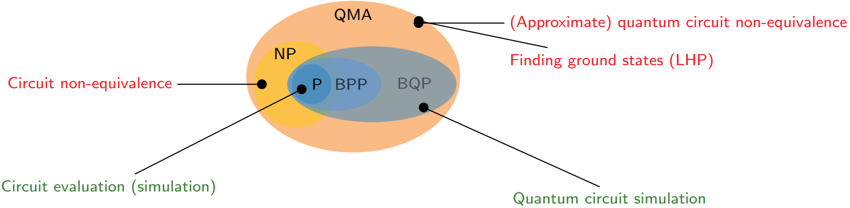
L	CD	FO	SFO	∧C	∧BC	∨C	∨BC	¬C
NNF	√	∘	√	√	√	√	√	√
DNNF	√	√	√	∘	∘	√	√	∘
d-NNF	√	∘	√	√	√	√	√	√
s-NNF	√	∘	√	√	√	√	√	√
f-NNF	√	∘	√	•	•	•	•	√
d-DNNF	√	∘	∘	∘	∘	∘	∘	?
sd-DNNF	√	∘	∘	∘	∘	∘	∘	?
BDD	√	∘	√	√	√	√	√	√
FBDD	√	•	∘	•	∘	•	∘	√
OBDD	√	•	√	•	∘	•	∘	√
$\text{OBDD}_<$	√	•	√	•	√	•	√	√
DNF	√	√	√	•	√	√	√	•
CNF	√	∘	√	√	√	•	√	•
PI	√	√	√	•	•	•	√	•
IP	√	•	•	•	√	•	•	•
MODS	√	√	√	•	√	•	•	•

operation tractability

[1] A. Darwiche and P. Marquis, "A knowledge compilation map," *Journal of Artificial Intelligence Research*, vol. 17, 2002.



Broader Motivation: Knowledge Compilation for Quantum Information (BQP, QMA, BPP)



Broader Motivation: Knowledge Compilation for Quantum Information (BQP, QMA, BPP)

Initial Motivation: Shortcomings in the State-of-the-Art

Classical Representations of Quantum States

From Physics:

- Stabilizer formalism [1] ✓entanglement / ✗structure

[1] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits," *Physical Review A*, vol. 70, 2004.

[2] C. J. Wood, J. D. Biamonte, and D. G. Cory, "Tensor networks and graphical calculus for open quantum systems," *Quantum Info. Comput.*, vol. 15, 2015.

[3] G. Viamontes, I. Markov, and J. Hayes, "High-performance QuIDD-based simulation of quantum circuits," in *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, vol. 2, 2004.

[4] D. M. Miller and M. A. Thornton, "QMDD: A decision diagram structure for reversible and quantum circuits," *ISMVL*, 2006.

Classical Representations of Quantum States

From Physics:

- Stabilizer formalism [1] ✓entanglement / ✗structure
- Tensor networks [2] ✗entanglement / ✓structure

[1] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits," *Physical Review A*, vol. 70, 2004.

[2] C. J. Wood, J. D. Biamonte, and D. G. Cory, "Tensor networks and graphical calculus for open quantum systems," *Quantum Info. Comput.*, vol. 15, 2015.

[3] G. Viamontes, I. Markov, and J. Hayes, "High-performance QuIDD-based simulation of quantum circuits," in *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, vol. 2, 2004.

[4] D. M. Miller and M. A. Thornton, "QMDD: A decision diagram structure for reversible and quantum circuits," *ISMVL*, 2006.

Classical Representations of Quantum States

From Physics:

- Stabilizer formalism [1] ✓entanglement / ✗structure
- Tensor networks [2] ✗entanglement / ✓structure

From Computer Science:

- Decision Diagrams [3, 4] ?entanglement / ✓structure

[1] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits," *Physical Review A*, vol. 70, 2004.

[2] C. J. Wood, J. D. Biamonte, and D. G. Cory, "Tensor networks and graphical calculus for open quantum systems," *Quantum Info. Comput.*, vol. 15, 2015.

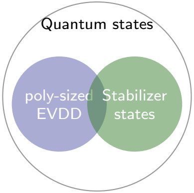
[3] G. Viamontes, I. Markov, and J. Hayes, "High-performance QuIDD-based simulation of quantum circuits," in *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, vol. 2, 2004.

[4] D. M. Miller and M. A. Thornton, "QMDD: A decision diagram structure for reversible and quantum circuits," *ISMVL*, 2006.

Entanglement in Decision Diagrams

EVDD can't succinctly represent stabilizer states, yet:

- Stabilizer circuits are classically simulatable! [1]
- Important in error correction, measurement-based QC, and circuit optimization.



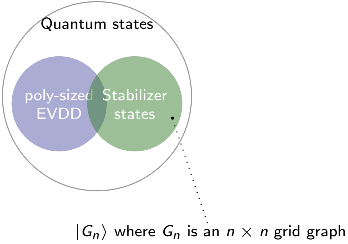
[1] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits," *Physical Review A*, vol. 70, 2004.

[2] L. Vinkhuijzen, T. Coopmans, D. Elkouss, V. Dunjko, and A. Laarman, "LIMDD: A decision diagram for simulation of quantum computing including stabilizer states," *Quantum*, vol. 7, 2023.

Entanglement in Decision Diagrams

EVDD can't succinctly represent stabilizer states, yet:

- Stabilizer circuits are classically simulatable! [1]
- Important in error correction, measurement-based QC, and circuit optimization.



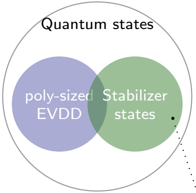
[1] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits," *Physical Review A*, vol. 70, 2004.

[2] L. Vinkhuijzen, T. Coopmans, D. Elkouss, V. Dunjko, and A. Laarman, "LIMDD: A decision diagram for simulation of quantum computing including stabilizer states," *Quantum*, vol. 7, 2023.

Entanglement in Decision Diagrams

EVDD can't succinctly represent stabilizer states, yet:

- Stabilizer circuits are classically simulatable! [1]
- Important in error correction, measurement-based QC, and circuit optimization.



$|G_n\rangle$ where G_n is an $n \times n$ grid graph

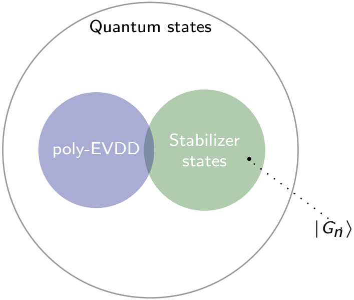
Theorem ([2])

EVDD requires $2^{\Omega(n)}$ nodes to represent $n \times n$ grid graph states.

[1] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits," *Physical Review A*, vol. 70, 2004.

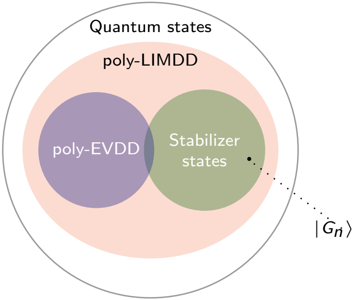
[2] L. Vinkhuijzen, T. Coopmans, D. Elkouss, V. Dunjko, and A. Laarman, "LIMDD: A decision diagram for simulation of quantum computing including stabilizer states," *Quantum*, vol. 7, 2023.

LIMDD Results



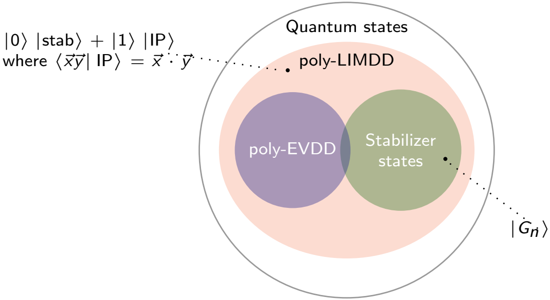
[1] L. Vinkhuijzen, T. Coopmans, and A. Laarman, "A knowledge compilation map for quantum information," in *AAAI*, vol. 28, 2026.

LIMDD Results



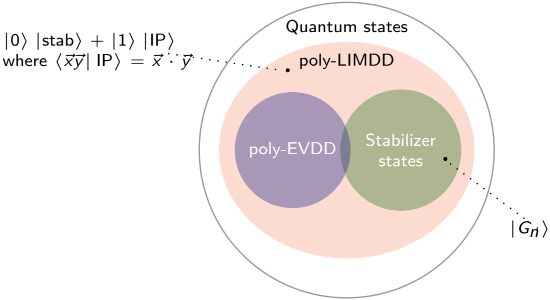
[1] L. Vinkhuijzen, T. Coopmans, and A. Laarman, "A knowledge compilation map for quantum information," in *AAAI*, vol. 28, 2026.

LIMDD Results



[1] L. Vinkhuijzen, T. Coopmans, and A. Laarman, "A knowledge compilation map for quantum information," in *AAAI*, vol. 28, 2026.

LIMDD Results



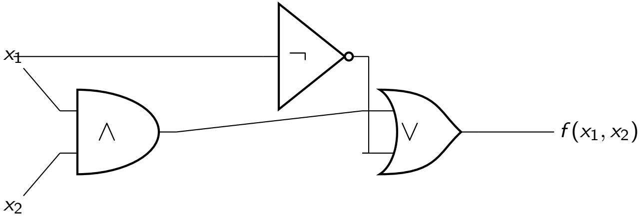
Operation \ input:	**EVDD**	**LIMDD**
Single-qubit Measurement	$\mathcal{O}(m)$	$\mathcal{O}(m)$
Single Pauli gate	$\mathcal{O}(m)$	$\mathcal{O}(1)$
Single Hadamard gate	$\mathcal{O}(2^n)$	$\mathcal{O}(n^3 2^n)$
Clifford gate on stabilizer state	$\mathcal{O}(2^n)$	$\mathcal{O}(n^4)$
Multi-qubit gate	$\mathcal{O}(4^n)$	$\mathcal{O}(n^3 4^n)$
Checking state equality	$\mathcal{O}(1)$	$\mathcal{O}(n^3)$

[1] L. Vinkhuijzen, T. Coopmans, and A. Laarman, "A knowledge compilation map for quantum information," in *AAAI*, vol. 28, 2026.

Overview

- **Background**
  - Decision Diagrams
  - Stabilizer formalism
- **Knowledge Representation for QI**
- **Conclusion**

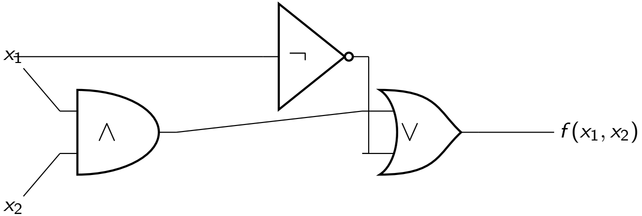
Boolean Circuits and Functions



Definition (Boolean Syntax)

$e \quad ::= \quad x \mid 0 \mid 1 \mid \neg e \mid e_1 \vee e_2 \mid e_1 \wedge e_2 \mid e_1 \implies e_2 \mid e_1 \Leftrightarrow e_2$

Boolean Circuits and Functions



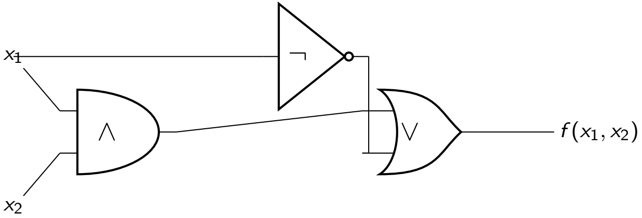
Definition (Boolean Syntax)

$e \quad ::= \quad x \mid 0 \mid 1 \mid \neg e \mid e_1 \vee e_2 \mid e_1 \wedge e_2 \mid e_1 \implies e_2 \mid e_1 \Leftrightarrow e_2$

Definition (Boolean Function)

$f : \mathbb{B}^n \rightarrow \mathbb{B} \quad \text{with} \quad \mathbb{B} \triangleq \{0, 1\}$

Boolean Circuits and Functions



Definition (Boolean Syntax)

$$e \quad ::= \quad x \mid 0 \mid 1 \mid \neg e \mid e_1 \vee e_2 \mid e_1 \wedge e_2 \mid e_1 \implies e_2 \mid e_1 \Leftrightarrow e_2$$

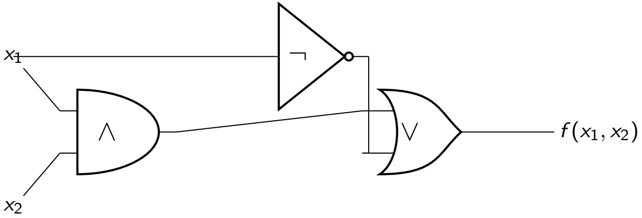
Definition (Boolean Function)

$$f : \mathbb{B}^n \to \mathbb{B} \quad \text{with} \quad \mathbb{B} \triangleq \{0, 1\}$$

Definition (Function restriction)

$$f_{|x_i:=b}(x_1, ..., x_i, ..., x_n) \triangleq f(x_1, .., b, ..., x_n)$$

Boolean Circuits and Functions



Definition (Boolean Syntax)

$e \quad ::= \quad x \mid 0 \mid 1 \mid \neg e \mid e_1 \vee e_2 \mid e_1 \wedge e_2 \mid e_1 \implies e_2 \mid e_1 \Leftrightarrow e_2$

Definition (Boolean Function)

$f : \mathbb{B}^n \rightarrow \mathbb{B} \quad$ with $\quad \mathbb{B} \triangleq \{0, 1\}$

Definition (Function restriction)

$f_{|x_i:=b}(x_1, ..., x_i, ..., x_n) \triangleq f(x_1, .., b, ..., x_n)$

$f_x := f_{x:=1}$
$f_{\overline{x}} := f_{x:=0}$

Shannon Expansion

Definition (Shannon Expansion)

$$f(x_1, x_2, \ldots, x_n) \iff x \wedge f(1, x_2, \ldots, x_n) \quad \vee \quad \overline{x} \wedge f(0, x_2, \ldots, x_n)$$

$$f \iff x\, f_{|x} \quad \vee \quad \overline{x}\, f_{|\overline{x}}$$

Theorem (Shannon Normal Form)

Let $F_{\vec{x}}$ be the recursive application of the Shannon expansion on any Boolean function f over variables X according to a total order $\vec{x} \in X^*$.
$F_{\vec{x}}$ is a canonical normal form.

Shannon Decomposition

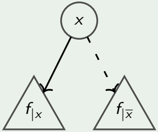
Example

$$
\begin{aligned}
(x_1 \wedge x_2) \vee \overline{x_3} \quad &\Longleftrightarrow \quad x_1(x_2 \vee \overline{x_3}) \vee \overline{x_1}(\overline{x_3}) && \text{(on } x_1\text{)} \\
(x_2 \vee \overline{x_3}) \quad &\Longleftrightarrow \quad x_2 \vee \overline{x_2 x_3} && \text{(on } x_2\text{)}
\end{aligned}
$$

Shannon Decomposition

Example

$$(x_1 \wedge x_2) \vee \overline{x_3} \iff x_1(x_2 \vee \overline{x_3}) \vee \overline{x_1}(\overline{x_3}) \qquad \text{(on } x_1\text{)}$$
$$(x_2 \vee \overline{x_3}) \iff x_2 \vee \overline{x_2 x_3} \qquad \text{(on } x_2\text{)}$$

Notation: 

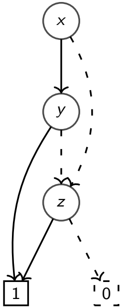
(a low edge and a high edge)

Read as:

- If x then $f_{|x}$ else $f_{|\overline{x}}$ (ITE)
- $(x \wedge f_{|x}) \vee (\overline{x} \wedge f_{|\overline{x}})$ (aka Shannon's expansion)

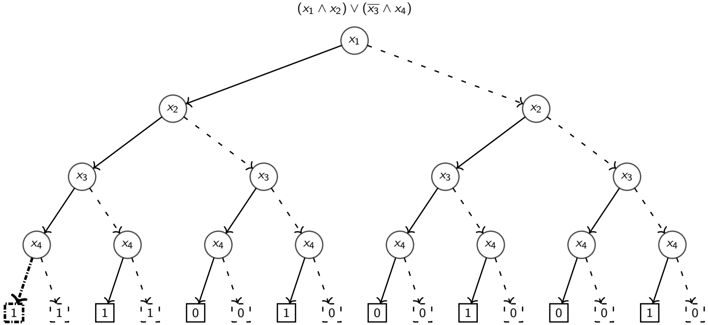
Binary Decision Diagrams (BDDs)

$$f(x, y, z) = (x \wedge y) \vee z$$



[1] R. E. Bryant, “Graph-based algorithms for Boolean function manipulation,” *IEEE Trans. Computers*, vol. 35, 1986.

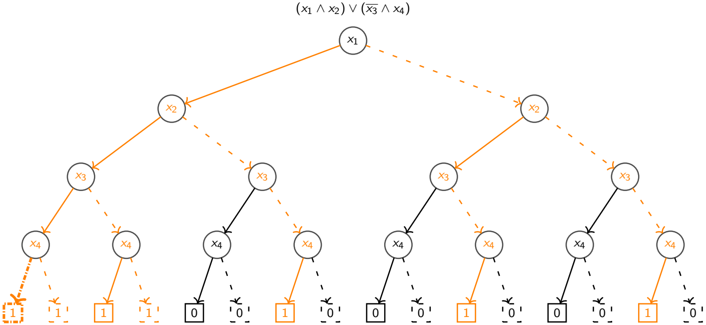
Ordered Binary Decision Diagram (BDD)



Decision Diagram

- Internal nodes are of type N: (X, N, N)
- View as set: $\{\,(1111)\,,(1110)\,,(1101)\,,(1100)\,,(1001)\,,(0101)\,,(0001)\,\}$

Ordered Binary Decision Diagram (BDD)



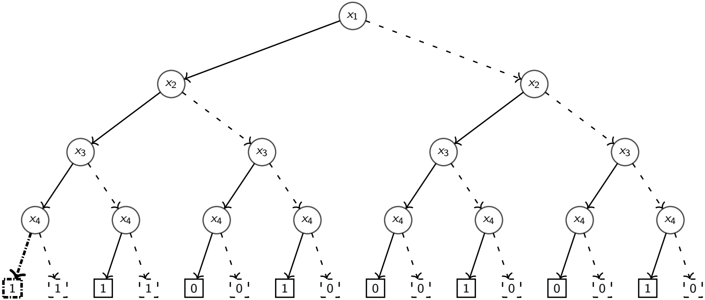
Decision Diagram

- Internal nodes are of type N: (X, N, N)
- View as set: $\{ (1111), (1110), (1101), (1100), (1001), (0101), (0001) \}$

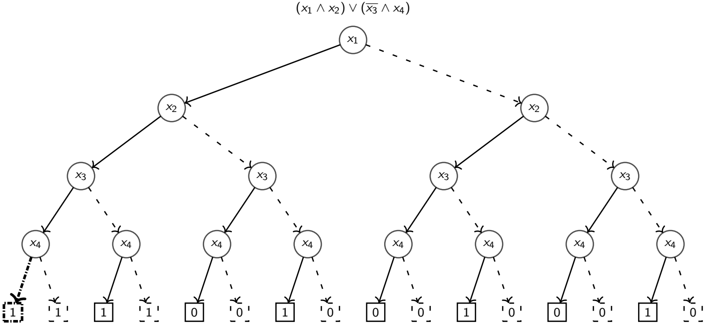
A BDD is a reduced decision tree

Reduced Ordered Binary Decision Diagram (BDD)

$(x_1 \wedge x_2) \vee (\overline{x_3} \wedge x_4)$



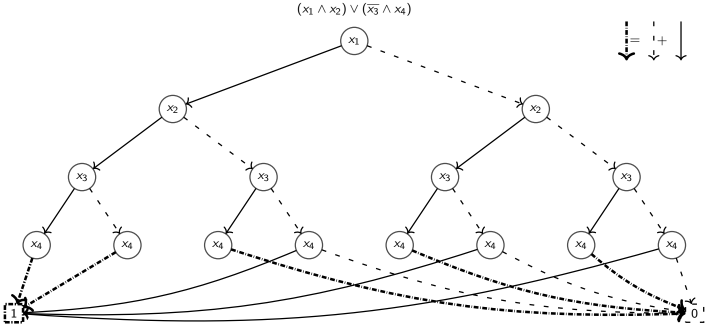
Reduced Ordered Binary Decision Diagram (BDD)



Reduced Binary Decision Diagrams

MERGE Merge isomorphic nodes $N\colon\ (X, N, N)$

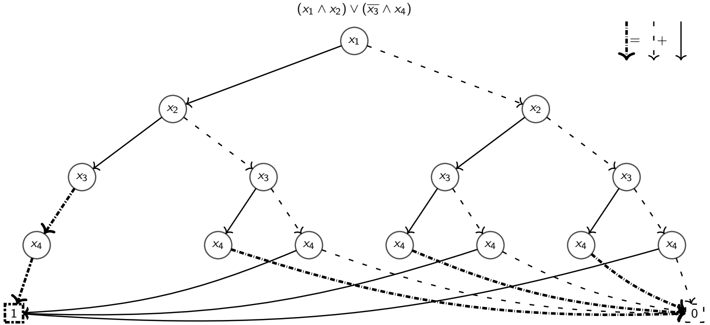
Reduced Ordered Binary Decision Diagram (BDD)



Reduced Binary Decision Diagrams

MERGE Merge isomorphic nodes N: (X, N, N)

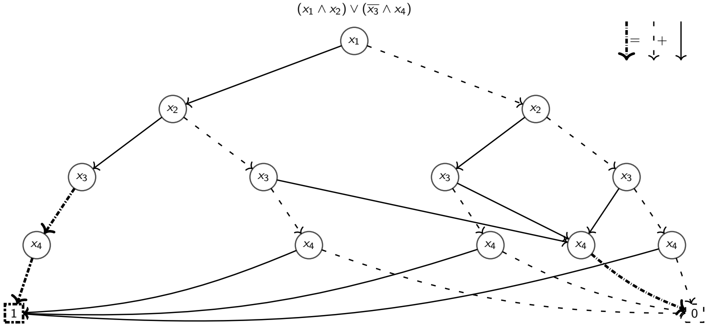
Reduced Ordered Binary Decision Diagram (BDD)



Reduced Binary Decision Diagrams

MERGE Merge isomorphic nodes N: (X, N, N)

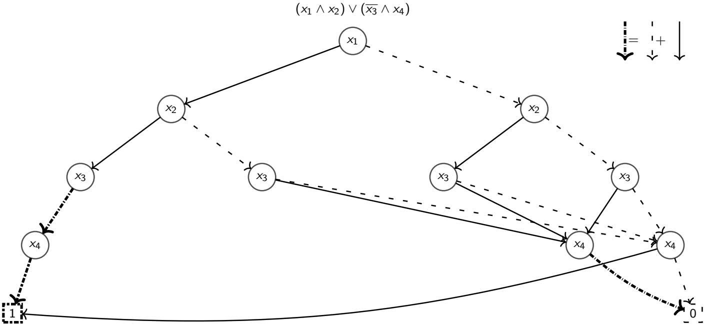
Reduced Ordered Binary Decision Diagram (BDD)



Reduced Binary Decision Diagrams

MERGE Merge isomorphic nodes N: (X, N, N)

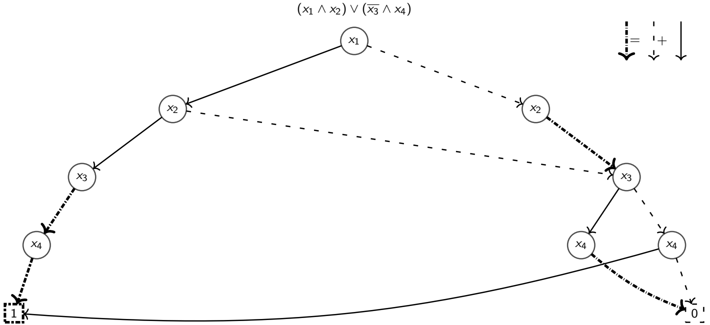
Reduced Ordered Binary Decision Diagram (BDD)



Reduced Binary Decision Diagrams

MERGE Merge isomorphic nodes N: (X, N, N)

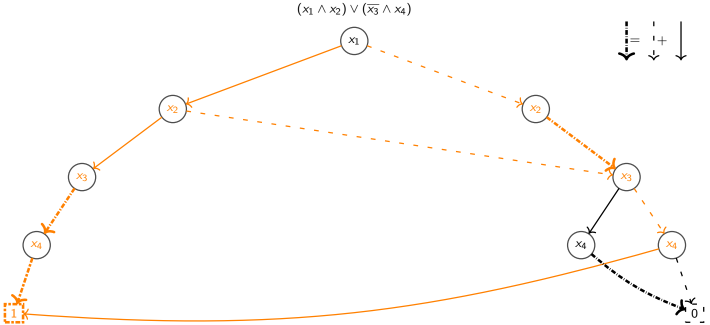
Reduced Ordered Binary Decision Diagram (BDD)



Reduced Binary Decision Diagrams

MERGE Merge isomorphic nodes N: (X, N, N)

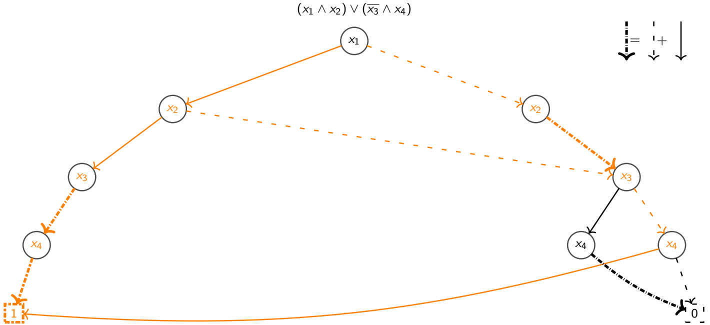
Reduced Ordered Binary Decision Diagram (BDD)



Reduced Binary Decision Diagrams

MERGE Merge isomorphic nodes N: (X, N, N)

Reduced Ordered Binary Decision Diagram (BDD)



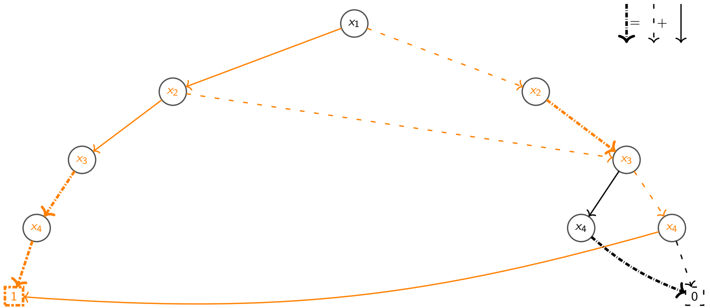
Reduced Binary Decision Diagrams

MERGE Merge isomorphic nodes N: (X, N, N)

How to implement?

Reduced Ordered Binary Decision Diagram (BDD)

$(x_1 \wedge x_2) \vee (\overline{x_3} \wedge x_4)$



Reduced Binary Decision Diagrams

MERGE Merge isomorphic nodes N: (X, N, N)

How to implement? Bottom up hashing of nodes N: (X, N, N)!

Reduced Ordered Binary Decision Diagram (BDD)

Definition (Extension of a Boolean function)

$[\![f]\!] = \{\, \vec{b} \in \{\, 0, 1 \,\}^n \mid f(\vec{b}) = 1 \,\}$

Reduced Ordered Binary Decision Diagram (BDD)

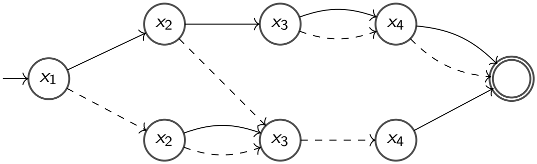
Definition (Extension of a Boolean function)

$[\![f]\!] = \{ \vec{b} \in \{ 0, 1 \}^n \mid f(\vec{b}) = 1 \}$

Automata view

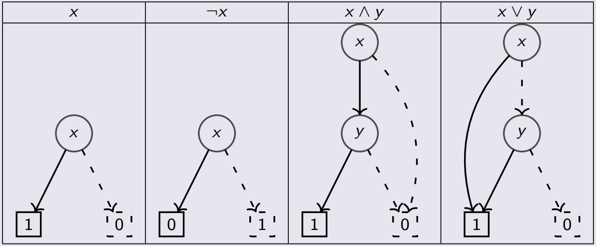
1. Fix the variable order.
2. View the extension as a language of fixed-length words: $[\![f]\!] \subseteq \{ 0, 1 \}^n$.
3. Let A_f be an automaton such that $\mathcal{L}(A_f) = [\![f]\!]$.
4. Automata reduction on A_f yields an ROBDD (without 0 leaf).

$(x_1 \wedge x_2) \vee (\overline{x_3} \wedge x_4)$



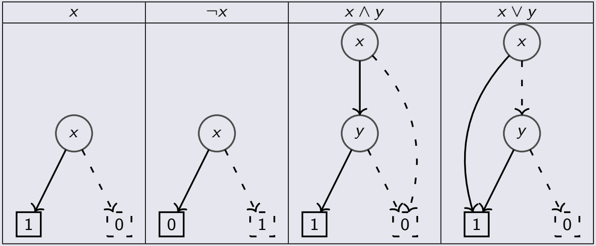
Examples

Basic Boolean Connectives



Examples

Basic Boolean Connectives

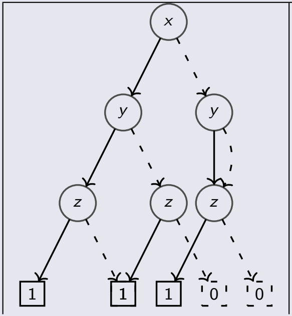


Propositional logic formulas

- Apparently, BDDs form an alternative to proposition logic.
- Recall negation $\neg$ and the binary connectives: $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$
- Which subset of operators is universal? [SHANNON]
- Introduce a ternary operator: $ITE(x, B_1, B_2)$; . . . . universal basis!

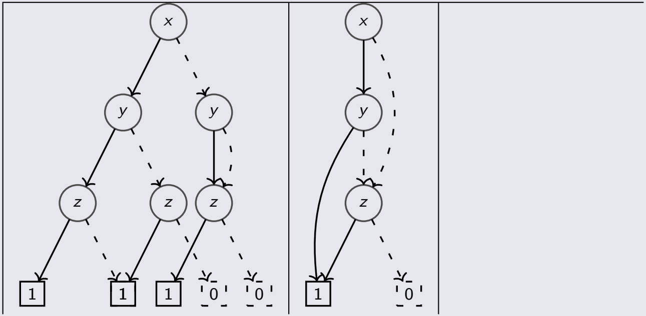
More Examples

Three times: $(x \wedge y) \vee z$



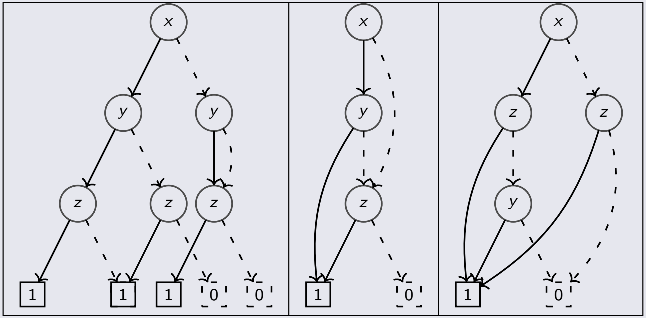
More Examples

Three times: $(x \wedge y) \vee z$



More Examples

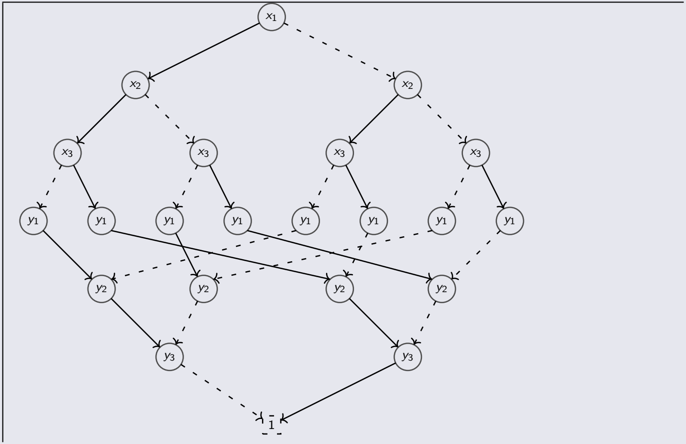
Three times: $(x \wedge y) \vee z$



Variable ordering can make exponential difference

$(x_1 \Leftrightarrow y_1) \wedge (x_2 \Leftrightarrow y_2) \wedge (x_3 \Leftrightarrow y_3)$ (edges to 0 are suppressed)

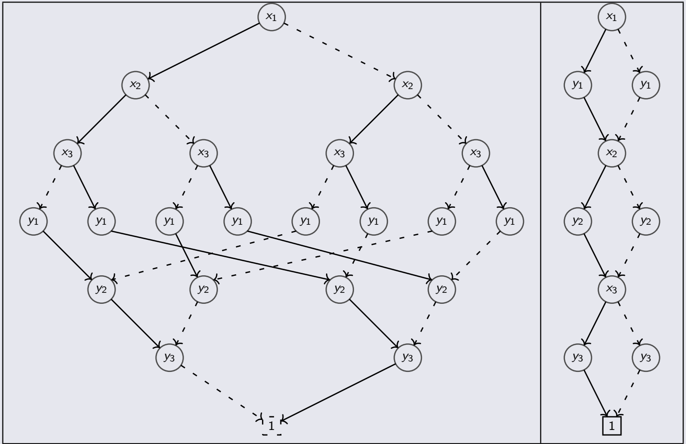
$x_1 < x_2 < x_3 < y_1 < y_2 < y_3$ .... $x_1 < y_1 < x_2 < y_2 < x_3 < y_3$



Variable ordering can make exponential difference

$(x_1 \Leftrightarrow y_1) \wedge (x_2 \Leftrightarrow y_2) \wedge (x_3 \Leftrightarrow y_3)$ (edges to 0 are suppressed)

$x_1 < x_2 < x_3 < y_1 < y_2 < y_3$ .... $x_1 < y_1 < x_2 < y_2 < x_3 < y_3$



$\mathcal{O}(2^n)$ nodes | $\mathcal{O}(n)$ nodes

Theoretical Results

Existence and Uniqueness

For a fixed variable ordering $(\vec{x}, <)$, every Boolean function can be represented by a canonical ROBDD.

Theoretical Results

Existence and Uniqueness

For a fixed variable ordering $(\vec{x}, <)$, every Boolean function can be represented by a canonical ROBDD.

Variable ordering has a huge impact on the OBDD size

- Finding the optimal ordering is NP-hard
- Some functions only admit exponentially large OBDDs
  - E.g.: multiplication $P(\vec{x}, \vec{y}, \vec{z})$ such that

$$(x_1 \ldots x_n) * (y_1 \ldots y_n) = (z_1 \ldots z_{2n})$$

    needs $\emptyset(2^n)$ OBDD nodes, whatever ordering is chosen
- Recall that almost all Boolean functions have exponential circuits [SHANNON 1949]
- Many practical functions have small OBDD representations!

BDD Operations

Recursive Formulation

Z = AND(X,Y)

$$Z \iff xZ_{|x} \vee \overline{x}Z_{|\overline{x}} \qquad \text{(Shannon Expansion)}$$

BDD Operations

Recursive Formulation

Z = AND(X,Y)

$$Z \iff xZ_{|x} \vee \overline{x}Z_{|\overline{x}} \qquad \text{(Shannon Expansion)}$$

$$\iff x(X \wedge Y)_{|x} \vee \overline{x}(X \wedge Y)_{|\overline{x}} \qquad \text{(replace Z)}$$

BDD Operations

Recursive Formulation

Z = AND(X,Y)

$$
\begin{aligned}
Z &\iff xZ_{|x} \vee \overline{x}Z_{|\overline{x}} && \text{(Shannon Expansion)} \\
&\iff x(X \wedge Y)_{|x} \vee \overline{x}(X \wedge Y)_{|\overline{x}} && \text{(replace Z)} \\
&\iff x(\underbrace{X_{|x} \wedge Y_{|x}}) \vee \overline{x}(\underbrace{X_{|\overline{x}} \wedge Y_{|\overline{x}}}) && \text{(distribute restriction)}
\end{aligned}
$$

BDD Operations

Recursive Formulation

Z = AND(X,Y)

$$Z \iff xZ_{|x} \vee \overline{x}Z_{|\overline{x}} \quad \text{(Shannon Expansion)}$$

$$\iff x(X \wedge Y)_{|x} \vee \overline{x}(X \wedge Y)_{|\overline{x}} \quad \text{(replace Z)}$$

$$\iff x(\underbrace{X_{|x} \wedge Y_{|x}}_{\text{recursive!}}) \vee \overline{x}(\underbrace{X_{|\overline{x}} \wedge Y_{|\overline{x}}}_{\text{recursive!}}) \quad \text{(distribute restriction)}$$

BDD Operations

Recursive Formulation

Z = AND(X,Y)

$$Z \iff x Z_{|x} \vee \overline{x} Z_{|\overline{x}} \qquad \text{(Shannon Expansion)}$$

$$\iff x(X \wedge Y)_{|x} \vee \overline{x}(X \wedge Y)_{|\overline{x}} \qquad \text{(replace Z)}$$

$$\iff x(\underbrace{X_{|x} \wedge Y_{|x}}_{\text{recursive!}}) \vee \overline{x}(\underbrace{X_{|\overline{x}} \wedge Y_{|\overline{x}}}_{\text{recursive!}}) \qquad \text{(distribute restriction)}$$

$$\iff \mathrm{ITE}(x,\ \underbrace{X_{|x} \wedge Y_{|x}}_{\text{recursive!}},\ \underbrace{X_{|\overline{x}} \wedge Y_{|\overline{x}}}_{\text{recursive!}}) \qquad \text{(this is a BDD node!)}$$

Remarks

BDD Operations

Recursive Formulation

Z = AND(X,Y)

$$
\begin{aligned}
Z &\iff xZ_{|x} \vee \overline{x}Z_{|\overline{x}} && \text{(Shannon Expansion)}\\
&\iff x(X \wedge Y)_{|x} \vee \overline{x}(X \wedge Y)_{|\overline{x}} && \text{(replace Z)}\\
&\iff x(\underbrace{X_{|x} \wedge Y_{|x}}_{\text{recursive!}}) \vee \overline{x}(\underbrace{X_{|\overline{x}} \wedge Y_{|\overline{x}}}_{\text{recursive!}}) && \text{(distribute restriction)}\\
&\iff \mathrm{ITE}(x,\ \underbrace{X_{|x} \wedge Y_{|x}}_{\text{recursive!}},\ \underbrace{X_{|\overline{x}} \wedge Y_{|\overline{x}}}_{\text{recursive!}}) && \text{(this is a BDD node!)}
\end{aligned}
$$

Remarks

- With dynamic programming, the AND(A,B)-operation runs in linear time in the number of BDD nodes $|A| \times |B|$ (also for: OR, NOT, $\exists x_1$, etc).

BDD Operations

Recursive Formulation

Z = AND(X,Y)

$$
\begin{aligned}
Z &\iff xZ_{|x} \lor \overline{x}Z_{|\overline{x}} && \text{(Shannon Expansion)}\\
&\iff x(X \land Y)_{|x} \lor \overline{x}(X \land Y)_{|\overline{x}} && \text{(replace Z)}\\
&\iff x(\underbrace{X_{|x} \land Y_{|x}}_{\text{recursive!}}) \lor \overline{x}(\underbrace{X_{|\overline{x}} \land Y_{|\overline{x}}}_{\text{recursive!}}) && \text{(distribute restriction)}\\
&\iff \mathrm{ITE}(x,\ \underbrace{X_{|x} \land Y_{|x}}_{\text{recursive!}},\ \underbrace{X_{|\overline{x}} \land Y_{|\overline{x}}}_{\text{recursive!}}) && \text{(this is a BDD node!)}
\end{aligned}
$$

Remarks

- With dynamic programming, the AND(A,B)-operation runs in linear time in the number of BDD nodes $|A| \times |B|$ (also for: OR, NOT, $\exists x_1$, etc).
- Operations build BDDs bottom-up, so we can reduce immediately (by hashing).

BDD Operations

Recursive Formulation

Z = AND(X,Y)

$$
\begin{aligned}
Z &\iff xZ_{|x} \vee \overline{x}Z_{|\overline{x}} && \text{(Shannon Expansion)} \\
&\iff x(X \wedge Y)_{|x} \vee \overline{x}(X \wedge Y)_{|\overline{x}} && \text{(replace Z)} \\
&\iff x(\underbrace{X_{|x} \wedge Y_{|x}}_{\text{recursive!}}) \vee \overline{x}(\underbrace{X_{|\overline{x}} \wedge Y_{|\overline{x}}}_{\text{recursive!}}) && \text{(distribute restriction)} \\
&\iff \mathrm{ITE}(x, \underbrace{X_{|x} \wedge Y_{|x}}_{\text{recursive!}}, \underbrace{X_{|\overline{x}} \wedge Y_{|\overline{x}}}_{\text{recursive!}}) && \text{(this is a BDD node!)}
\end{aligned}
$$

Remarks

- With dynamic programming, the AND(A,B)-operation runs in linear time in the number of BDD nodes $|A| \times |B|$ (also for: OR, NOT, $\exists x_1$, etc).
- Operations build BDDs bottom-up, so we can reduce immediately (by hashing).
- We have "Homomorphic Compression".

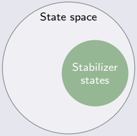
Overview

- **Background**
  - Decision Diagrams
  - Stabilizer formalism
- **Knowledge Representation for QI**
- **Conclusion**

Quantum State Representation with the Stabilizer Formalism

Definition (Stabilizer States [1])

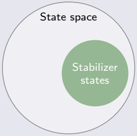
- A non-universal subset of all quantum states crucial for error correction, etc:



Quantum State Representation with the Stabilizer Formalism

Definition (Stabilizer States [1])

- A non-universal subset of all quantum states crucial for error correction, etc:



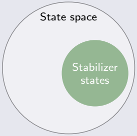
- A stabilizer state $|\varphi\rangle$ is uniquely defined by n-qubit Pauli strings ($n\times$):

$$\left\{\begin{array}{ccccccc} \pm P_{1,1} & \otimes & P_{1,2} & \otimes & \dots & \otimes & P_{1,n}, \\ \pm P_{2,1} & \otimes & P_{2,2} & \otimes & \dots & \otimes & P_{2,n}, \\ \vdots & & \vdots & & \ddots & & \vdots \\ \pm P_{n,1} & \otimes & P_{n,2} & \otimes & \dots & \otimes & P_{n,n}, \end{array}\right\} \text{ for } P_{i,j} \in \underbrace{\{\mathbb{I}, X, Y, Z\}}_{\text{single qubit Pauli's}}$$

Quantum State Representation with the Stabilizer Formalism

Definition (Stabilizer States [1])

- A non-universal subset of all quantum states crucial for error correction, etc:



- A stabilizer state $|\varphi\rangle$ is uniquely defined by n-qubit Pauli strings ($n\times$):

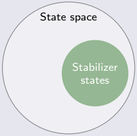
$$\left\{\begin{array}{ccccccc} \pm P_{1,1} & \otimes & P_{1,2} & \otimes & \dots & \otimes & P_{1,n}, \\ \pm P_{2,1} & \otimes & P_{2,2} & \otimes & \dots & \otimes & P_{2,n}, \\ \vdots & & \vdots & & \ddots & & \vdots \\ \pm P_{n,1} & \otimes & P_{n,2} & \otimes & \dots & \otimes & P_{n,n}, \end{array}\right\} \text{ for } P_{i,j} \in \underbrace{\{\mathbb{I}, X, Y, Z\}}_{\text{single qubit Pauli's}}$$

with $\pm P_{i,1} \otimes P_{i,2} \otimes \cdots \otimes P_{i,n} \cdot |\varphi\rangle = |\varphi\rangle$ (state $|\varphi\rangle$ is stabilized by each Pauli op.)

Quantum State Representation with the Stabilizer Formalism

Definition (Stabilizer States [1])

- A non-universal subset of all quantum states crucial for error correction, etc:



- A stabilizer state $|\varphi\rangle$ is uniquely defined by n-qubit Pauli strings ($n\times$):

$$\begin{Bmatrix} \pm P_{1,1} & \otimes & P_{1,2} & \otimes & \dots & \otimes & P_{1,n}, \\ \pm P_{2,1} & \otimes & P_{2,2} & \otimes & \dots & \otimes & P_{2,n}, \\ \vdots & & \vdots & & \ddots & & \vdots \\ \pm P_{n,1} & \otimes & P_{n,2} & \otimes & \dots & \otimes & P_{n,n}, \end{Bmatrix} \text{ for } P_{i,j} \in \underbrace{\{\,\mathbb{I}, X, Y, Z\,\}}_{\text{single qubit Pauli's}}$$

  with $\pm P_{i,1} \otimes P_{i,2} \otimes \cdots \otimes P_{i,n} \cdot |\varphi\rangle = |\varphi\rangle$ (state $|\varphi\rangle$ is stabilized by each Pauli op.)
- Closed under the stabilizer (or Clifford) gate set: $\{\,S, H, CX\,\}$
- Efficiently classically simulatable!

[1] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits," *Physical Review A*, vol. 70, 2004.

Quantum State Representation with the Stabilizer Formalism

Pauli gates: $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

Two-qubit stabilizer states

- $Z \cdot |0\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle$
- $-Z \cdot |1\rangle = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$

Quantum State Representation with the Stabilizer Formalism

Pauli gates: $I = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}, \quad X = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix} \quad Y = \begin{bmatrix}0 & -i\\i & 0\end{bmatrix} \quad Z = \begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}$

Two-qubit stabilizer states

- $Z \cdot |0\rangle = \begin{bmatrix}1 & 0\\0 & -1\end{bmatrix} \cdot \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}1\\0\end{bmatrix} = |0\rangle$
- $-Z \cdot |1\rangle = \begin{bmatrix}-1 & 0\\0 & 1\end{bmatrix} \cdot \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\1\end{bmatrix} = |1\rangle$
- $X \cdot |+\rangle = \quad X \cdot \begin{bmatrix}1\\1\end{bmatrix} = \begin{bmatrix}1\\1\end{bmatrix} = |+\rangle$, etc

Quantum State Representation with the Stabilizer Formalism

Pauli gates: $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

Two-qubit stabilizer states

- $Z \cdot |0\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle$
- $-Z \cdot |1\rangle = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$
- $X \cdot |+\rangle = \quad X \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} = |+\rangle$, etc
- The computational basis state $|00\rangle$ is stabilized by $\begin{Bmatrix} Z \otimes I, \\ I \otimes Z \end{Bmatrix}$
- The Bell state $|\Phi\rangle \approx |00\rangle + |11\rangle$ is stabilized by $\begin{Bmatrix} Z \otimes Z, \\ X \otimes X \end{Bmatrix}$,

Quantum State Representation with the Stabilizer Formalism

Pauli gates: $I = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$, $X = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$ $Y = \begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}$ $Z = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$

Two-qubit stabilizer states

- $Z \cdot |0\rangle = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} \cdot \begin{bmatrix}1\\ 0\end{bmatrix} = \begin{bmatrix}1\\ 0\end{bmatrix} = |0\rangle$
- $-Z \cdot |1\rangle = \begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix} \cdot \begin{bmatrix}0\\ 1\end{bmatrix} = \begin{bmatrix}0\\ 1\end{bmatrix} = |1\rangle$
- $X \cdot |+\rangle = \quad X \cdot \begin{bmatrix}1\\ 1\end{bmatrix} = \begin{bmatrix}1\\ 1\end{bmatrix} = |+\rangle$, etc
- The computational basis state $|00\rangle$ is stabilized by $\left\{\begin{matrix}Z \otimes I,\\ I \otimes Z\end{matrix}\right\}$
- The Bell state $|\Phi\rangle \approx |00\rangle + |11\rangle$ is stabilized by $\left\{\begin{matrix}Z \otimes Z,\\ X \otimes X\end{matrix}\right\}$, e.g., $X \otimes X \cdot |\Phi\rangle = |\Phi\rangle$

Quantum State Representation with the Stabilizer Formalism

Pauli gates: $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

Two-qubit stabilizer states

- $Z \cdot |0\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle$
- $-Z \cdot |1\rangle = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$
- $X \cdot |+\rangle = X \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} = |+\rangle$, etc
- The computational basis state $|00\rangle$ is stabilized by $\left\{ \begin{matrix} Z \otimes I, \\ I \otimes Z \end{matrix} \right\}$
- The Bell state $|\Phi\rangle \approx |00\rangle + |11\rangle$ is stabilized by $\left\{ \begin{matrix} Z \otimes Z, \\ X \otimes X \end{matrix} \right\}$, e.g., $X \otimes X \cdot |\Phi\rangle = |\Phi\rangle$

We can apply gate H to qubit two of $|\Phi\rangle$ as follows:

$$\left\{ \begin{matrix} Z \otimes Z, \\ X \otimes X \end{matrix} \right\} \xrightarrow{\quad I \otimes H \quad} \left\{ \begin{matrix} Z \otimes X, \\ X \otimes Z \end{matrix} \right\}$$

Quantum State Representation with the Stabilizer Formalism

Pauli gates: $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$

Two-qubit stabilizer states

- $Z \cdot |0\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle$
- $-Z \cdot |1\rangle = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle$
- $X \cdot |+\rangle = \ X \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} = |+\rangle$, etc
- The computational basis state $|00\rangle$ is stabilized by $\begin{Bmatrix} Z \otimes I, \\ I \otimes Z \end{Bmatrix}$
- The Bell state $|\Phi\rangle \approx |00\rangle + |11\rangle$ is stabilized by $\begin{Bmatrix} Z \otimes Z, \\ X \otimes X \end{Bmatrix}$, e.g., $X \otimes X \cdot |\Phi\rangle = |\Phi\rangle$

We can apply gate H to qubit two of $|\Phi\rangle$ as follows:

$$\begin{Bmatrix} Z \otimes Z, \\ X \otimes X \end{Bmatrix} \xrightarrow{\quad I \otimes H \quad} \begin{Bmatrix} Z \otimes X, \\ X \otimes Z \end{Bmatrix}$$

Quantum State Representation with the Stabilizer Formalism

Mathematical properties

Let $\mathcal{P}_n = i^c \cdot \{ I, X, Y, Z \}^{\otimes n}$ for $c \in \{ 0, 1, 2, 3 \}$ by the n-qubit Pauli group.

Observe that stabilizers form a group, because of P_1 and P_2 stabilize $|\varphi\rangle$, then so does $P_1 \cdot P_2 \cdot |\varphi\rangle = |\varphi\rangle$.

We have that:

Quantum State Representation with the Stabilizer Formalism

Mathematical properties

Let $\mathcal{P}_n = i^c \cdot \{ I, X, Y, Z \}^{\otimes n}$ for $c \in \{ 0, 1, 2, 3 \}$ by the n-qubit Pauli group.

Observe that stabilizers form a group, because of P_1 and P_2 stabilize $|\varphi\rangle$, then so does $P_1 \cdot P_2 \cdot |\varphi\rangle = |\varphi\rangle$.

We have that:

- A stabilizer state is stabilized by the largest subgroup of $\mathcal{P}_n$ that does not contain $-I^{\otimes n}$.

Quantum State Representation with the Stabilizer Formalism

Mathematical properties

Let $\mathcal{P}_n = i^c \cdot \{ I, X, Y, Z \}^{\otimes n}$ for $c \in \{ 0, 1, 2, 3 \}$ by the n-qubit Pauli group.

Observe that stabilizers form a group, because of P_1 and P_2 stabilize $|\varphi\rangle$, then so does $P_1 \cdot P_2 \cdot |\varphi\rangle = |\varphi\rangle$.

We have that:

- A stabilizer state is stabilized by the largest subgroup of $\mathcal{P}_n$ that does not contain $-I^{\otimes n}$.
- Such a group is abelian (commutative).

Quantum State Representation with the Stabilizer Formalism

Mathematical properties

Let $\mathcal{P}_n = i^c \cdot \{ I, X, Y, Z \}^{\otimes n}$ for $c \in \{ 0, 1, 2, 3 \}$ by the n-qubit Pauli group.

Observe that stabilizers form a group, because of P_1 and P_2 stabilize $|\varphi\rangle$, then so does $P_1 \cdot P_2 \cdot |\varphi\rangle = |\varphi\rangle$.

We have that:

- A stabilizer state is stabilized by the largest subgroup of $\mathcal{P}_n$ that does not contain $-I^{\otimes n}$.
- Such a group is abelian (commutative).
- Such a group only contains elements $\mathcal{P}'_n = \pm 1 \cdot \{ I, X, Y, Z \}^{\otimes n}$.

Quantum State Representation with the Stabilizer Formalism

Mathematical properties

Let $\mathcal{P}_n = i^c \cdot \{ I, X, Y, Z \}^{\otimes n}$ for $c \in \{ 0, 1, 2, 3 \}$ by the n-qubit Pauli group.

Observe that stabilizers form a group, because of P_1 and P_2 stabilize $|\varphi\rangle$, then so does $P_1 \cdot P_2 \cdot |\varphi\rangle = |\varphi\rangle$.

We have that:

- A stabilizer state is stabilized by the largest subgroup of $\mathcal{P}_n$ that does not contain $-I^{\otimes n}$.
- Such a group is abelian (commutative).
- Such a group only contains elements $\mathcal{P}'_n = \pm 1 \cdot \{ I, X, Y, Z \}^{\otimes n}$.
- Such a group has size 2^n.

Quantum State Representation with the Stabilizer Formalism

Mathematical properties

Let $\mathcal{P}_n = i^c \cdot \{ I, X, Y, Z \}^{\otimes n}$ for $c \in \{ 0, 1, 2, 3 \}$ by the n-qubit Pauli group.

Observe that stabilizers form a group, because of P_1 and P_2 stabilize $|\varphi\rangle$, then so does $P_1 \cdot P_2 \cdot |\varphi\rangle = |\varphi\rangle$.

We have that:

- A stabilizer state is stabilized by the largest subgroup of $\mathcal{P}_n$ that does not contain $-I^{\otimes n}$.
- Such a group is abelian (commutative).
- Such a group only contains elements $\mathcal{P}'_n = \pm 1 \cdot \{ I, X, Y, Z \}^{\otimes n}$.
- Such a group has size 2^n.
- Such a group is generated by some independent $G_1, \ldots, G_n \in \mathcal{P}'_n$ (the tableau).

Overview

- **Background**
- **Knowledge Representation for QI**
  - BDDs for Quantum States
  - LIMDD Definition
  - Quantum Knowledge Compilation Map
- **Conclusion**

Quantum State Representation with Decision Diagrams

We can think of a quantum state $\vec{v}$ as a pseudo Boolean function $f\colon \{0,1\}^n \to \mathbb{C}$, or

'Shannon decomposition' in quantum

$$\vec{v} = |0\rangle \otimes \underbrace{\vec{v}_0}_{n-1\text{ qubits}} + |1\rangle \otimes \underbrace{\vec{v}_1}_{n-1\text{ qubits}}$$

$$\vec{v} = \alpha_1 |00\ldots00\rangle + \cdots + \alpha_{2^n} |11\ldots11\rangle \qquad \text{(recursively)}$$

Quantum State Representation with Decision Diagrams

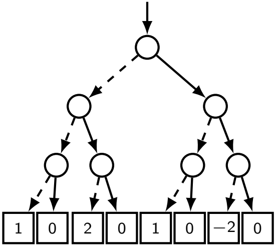
We can think of a quantum state $\vec{v}$ as a pseudo Boolean function $f\colon \{0,1\}^n \to \mathbb{C}$, or

'Shannon decomposition' in quantum

$$\vec{v} = |0\rangle \otimes \underbrace{\vec{v}_0}_{n-1 \text{ qubits}} \quad + \quad |1\rangle \otimes \underbrace{\vec{v}_1}_{n-1 \text{ qubits}}$$

$$\vec{v} = \alpha_1 |00\ldots00\rangle + \cdots + \alpha_{2^n} |11\ldots11\rangle \qquad \text{(recursively)}$$

$$\begin{bmatrix} 1 \\ 0 \\ 2 \\ 0 \\ 1 \\ 0 \\ -2 \\ 0 \end{bmatrix}$$



Decision tree

Quantum State Representation with Decision Diagrams

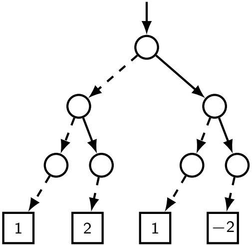
We can think of a quantum state $\vec{v}$ as a pseudo Boolean function $f\colon \{0,1\}^n \to \mathbb{C}$, or

'Shannon decomposition' in quantum

$$\vec{v} = |0\rangle \otimes \underbrace{\vec{v}_0}_{n-1 \text{ qubits}} \quad + \quad |1\rangle \otimes \underbrace{\vec{v}_1}_{n-1 \text{ qubits}}$$

$$\vec{v} = \alpha_1 |00\ldots00\rangle + \cdots + \alpha_{2^n} |11\ldots11\rangle \qquad \text{(recursively)}$$

$$\begin{bmatrix} 1 \\ 0 \\ 2 \\ 0 \\ 1 \\ 0 \\ -2 \\ 0 \end{bmatrix}$$



Decision tree

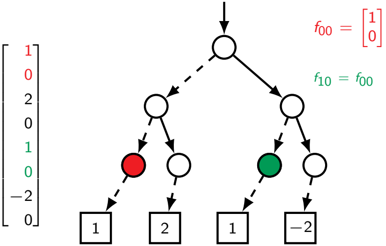
Quantum State Representation with Decision Diagrams

We can think of a quantum state $\vec{v}$ as a pseudo Boolean function $f\colon \{0,1\}^n \to \mathbb{C}$, or

'Shannon decomposition' in quantum

$$\vec{v} = |0\rangle \otimes \underbrace{\vec{v}_0}_{n-1 \text{ qubits}} \quad + \quad |1\rangle \otimes \underbrace{\vec{v}_1}_{n-1 \text{ qubits}}$$

$$\vec{v} = \alpha_1 |00\dots00\rangle + \dots + \alpha_{2^n} |11\dots11\rangle \qquad \text{(recursively)}$$



Decision tree

Quantum State Representation with Decision Diagrams

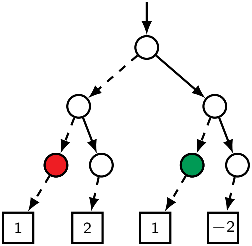
We can think of a quantum state $\vec{v}$ as a pseudo Boolean function $f\colon \{0,1\}^n \to \mathbb{C}$, or

'Shannon decomposition' in quantum

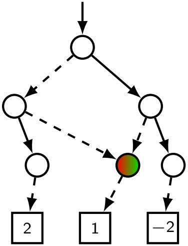
$$\vec{v} = |0\rangle \otimes \underbrace{\vec{v}_0}_{n-1 \text{ qubits}} \quad + \quad |1\rangle \otimes \underbrace{\vec{v}_1}_{n-1 \text{ qubits}}$$

$$\vec{v} = \alpha_1 |00\ldots00\rangle + \cdots + \alpha_{2^n} |11\ldots11\rangle \qquad \text{(recursively)}$$

$$\begin{bmatrix} 1 \\ 0 \\ 2 \\ 0 \\ 1 \\ 0 \\ -2 \\ 0 \end{bmatrix}$$



Decision tree



Multi-Terminal DD (MTBDD)

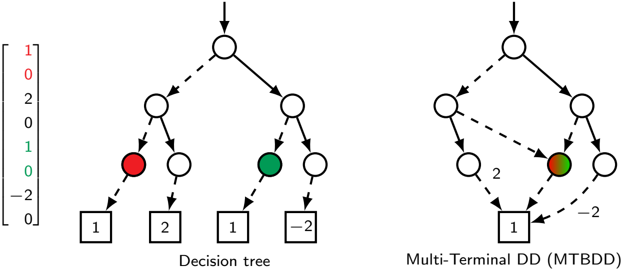
Quantum State Representation with Decision Diagrams

We can think of a quantum state $\vec{v}$ as a pseudo Boolean function $f\colon \{0,1\}^n \to \mathbb{C}$, or

'Shannon decomposition' in quantum

$$\vec{v} = |0\rangle \otimes \underbrace{\vec{v}_0}_{n-1\text{ qubits}} \quad + \quad |1\rangle \otimes \underbrace{\vec{v}_1}_{n-1\text{ qubits}}$$

$$\vec{v} = \alpha_1 |00\ldots00\rangle + \cdots + \alpha_{2^n} |11\ldots11\rangle \qquad \text{(recursively)}$$



Quantum State Representation with Decision Diagrams

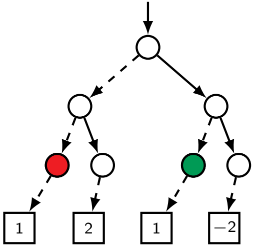
We can think of a quantum state $\vec{v}$ as a pseudo Boolean function $f\colon \{0,1\}^n \to \mathbb{C}$, or

'Shannon decomposition' in quantum

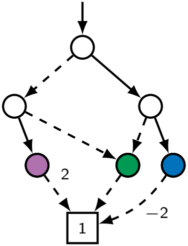
$$\vec{v} = |0\rangle \otimes \underbrace{\vec{v}_0}_{n-1 \text{ qubits}} \quad + \quad |1\rangle \otimes \underbrace{\vec{v}_1}_{n-1 \text{ qubits}}$$

$$\vec{v} = \alpha_1 |00\ldots00\rangle + \cdots + \alpha_{2^n} |11\ldots11\rangle \qquad \text{(recursively)}$$

$$\begin{bmatrix} 1 \\ 0 \\ 2 \\ 0 \\ 1 \\ 0 \\ -2 \\ 0 \end{bmatrix}$$



Decision tree



Multi-Terminal DD (MTBDD)

$f_{00} = f_{10} = \begin{bmatrix} 1, \\ 0 \end{bmatrix}$

$f_{01} = 2 \cdot f_{00}$

$f_{11} = -2 \cdot f_{00}$

Quantum State Representation with Decision Diagrams

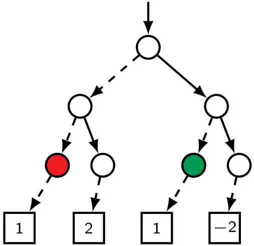
We can think of a quantum state $\vec{v}$ as a pseudo Boolean function $f\colon \{0,1\}^n \to \mathbb{C}$, or

'Shannon decomposition' in quantum

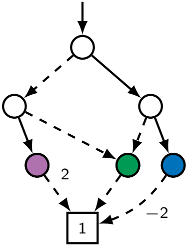
$$\vec{v} = |0\rangle \otimes \underbrace{\vec{v}_0}_{n-1\text{ qubits}} + |1\rangle \otimes \underbrace{\vec{v}_1}_{n-1\text{ qubits}}$$

$$\vec{v} = \alpha_1 |00\dots00\rangle + \dots + \alpha_{2^n} |11\dots11\rangle \qquad \text{(recursively)}$$

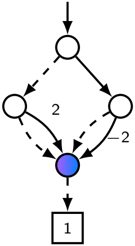
$$\begin{bmatrix} 1 \\ 0 \\ 2 \\ 0 \\ 1 \\ 0 \\ -2 \\ 0 \end{bmatrix}$$



Decision tree



Multi-Terminal DD (MTBDD)



Edge Valued DD (EVDD)

Quantum State Representation with Decision Diagrams

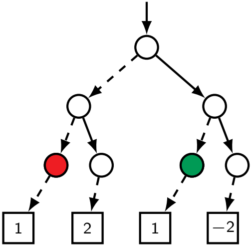
We can think of a quantum state $\vec{v}$ as a pseudo Boolean function $f\colon \{0,1\}^n \to \mathbb{C}$, or

'Shannon decomposition' in quantum

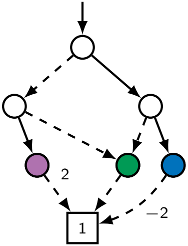
$$\vec{v} = |0\rangle \otimes \underbrace{\vec{v}_0}_{n-1\text{ qubits}} \quad + \quad |1\rangle \otimes \underbrace{\vec{v}_1}_{n-1\text{ qubits}}$$

$$\vec{v} = \alpha_1 |00\dots00\rangle + \cdots + \alpha_{2^n} |11\dots11\rangle \qquad \text{(recursively)}$$

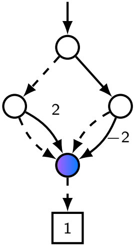
$$\begin{bmatrix} 1 \\ 0 \\ 2 \\ 0 \\ 1 \\ 0 \\ -2 \\ 0 \end{bmatrix}$$



Decision tree



Multi-Terminal DD (MTBDD)



Edge Valued DD (EVDD)

Merge nodes v, w: Never $\quad |v\rangle = |w\rangle \quad |v\rangle = \gamma \cdot |w\rangle$

(equivalent up to a complex factor γ)

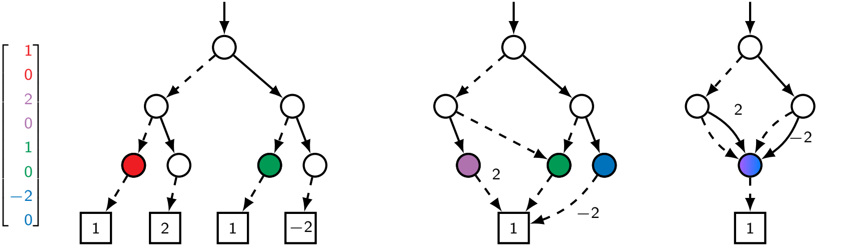
Quantum State Representation with Decision Diagrams

We can think of a quantum state $\vec{v}$ as a pseudo Boolean function $f\colon \{0,1\}^n \to \mathbb{C}$, or

'Shannon decomposition' in quantum

$$\vec{v} = |0\rangle \otimes \underbrace{\vec{v}_0}_{n-1 \text{ qubits}} \quad + \quad |1\rangle \otimes \underbrace{\vec{v}_1}_{n-1 \text{ qubits}}$$

$$\vec{v} = \alpha_1 |00\dots00\rangle + \dots + \alpha_{2^n} |11\dots11\rangle \qquad \text{(recursively)}$$



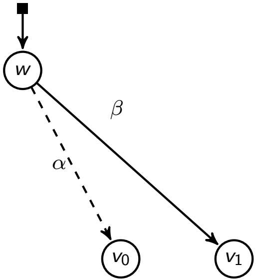
Decision tree | Multi-Terminal DD (MTBDD) | Edge Valued DD (EVDD)

Merge nodes v, w: Never | $|v\rangle = |w\rangle$ | $|v\rangle = \gamma \cdot |w\rangle$

(equivalent up to a complex factor γ)

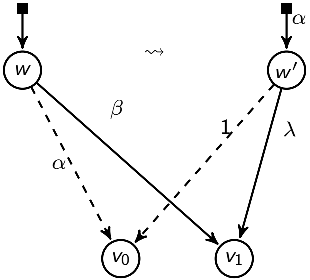
(EVDD can be exponentially more succinct than MTBDD.)

EVDD Canonicity



$\alpha, \beta \in \mathbb{C}$

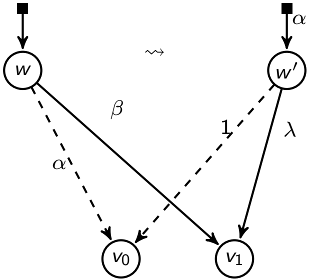
EVDD Canonicity



$\alpha, \beta \in \mathbb{C}$

$\lambda := \frac{\beta}{\alpha}$

EVDD Canonicity



$\alpha, \beta \in \mathbb{C}$

$\lambda := \frac{\beta}{\alpha}$

(This is a constant-time operation, which we can use to construct a *reduced* EVDD from the bottom up (and on-the-fly).)

Overview

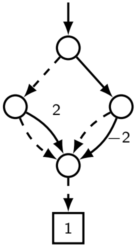
- **Background**
- **Knowledge Representation for QI**
  - BDDs for Quantum States
  - LIMDD Definition
  - Quantum Knowledge Compilation Map
- **Conclusion**

Pauli-Local Invertible Map DD (Pauli-LIMDD)

Idea: Instead of labeling edges with only constant factors $\alpha \in \mathbb{C}$, we use a constant factor and a Pauli string αP for $P \in \mathcal{P}_n$.

Pauli-Local Invertible Map DD (Pauli-LIMDD)

Idea: Instead of labeling edges with only constant factors $\alpha \in \mathbb{C}$, we use a constant factor and a Pauli string αP for $P \in \mathcal{P}_n$.

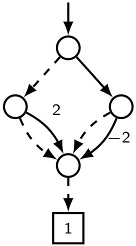


Edge-Valued DD (EVDD)

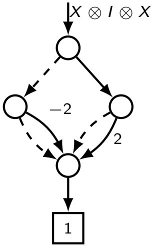
[1] L. Vinkhuijzen, T. Coopmans, D. Elkouss, V. Dunjko, and A. Laarman, "LIMDD: A decision diagram for simulation of quantum computing including stabilizer states," *Quantum*, vol. 7, 2023.

Pauli-Local Invertible Map DD (Pauli-LIMDD)

Idea: Instead of labeling edges with only constant factors $\alpha \in \mathbb{C}$, we use a constant factor and a Pauli string αP for $P \in \mathcal{P}_n$.



Edge-Valued DD (EVDD)

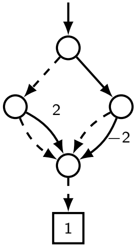


LIMDD [1]

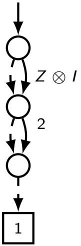
[1] L. Vinkhuijzen, T. Coopmans, D. Elkouss, V. Dunjko, and A. Laarman, "LIMDD: A decision diagram for simulation of quantum computing including stabilizer states," *Quantum*, vol. 7, 2023.

Pauli-Local Invertible Map DD (Pauli-LIMDD)

Idea: Instead of labeling edges with only constant factors $\alpha \in \mathbb{C}$, we use a constant factor and a Pauli string αP for $P \in \mathcal{P}_n$.



Edge-Valued DD (EVDD)

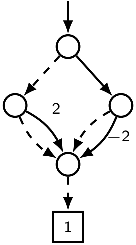


LIMDD [1]

[1] L. Vinkhuijzen, T. Coopmans, D. Elkouss, V. Dunjko, and A. Laarman, "LIMDD: A decision diagram for simulation of quantum computing including stabilizer states," *Quantum*, vol. 7, 2023.

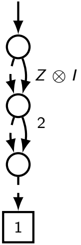
Pauli-Local Invertible Map DD (Pauli-LIMDD)

Idea: Instead of labeling edges with only constant factors $\alpha \in \mathbb{C}$, we use a constant factor and a Pauli string αP for $P \in \mathcal{P}_n$.



Edge-Valued DD (EVDD)

(merge nodes equivalent up to a constant factor)

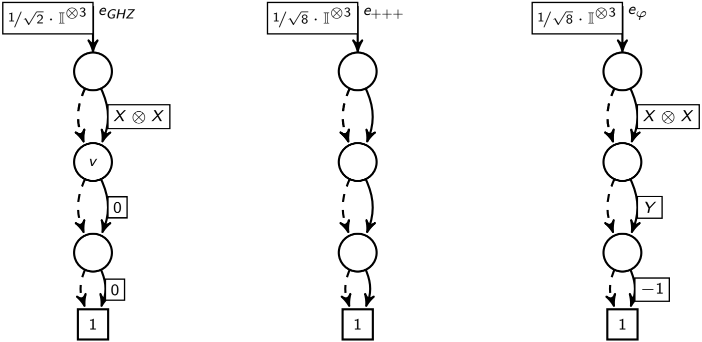


LIMDD [1]

(merge nodes equivalent up to stabilizer symmetry)

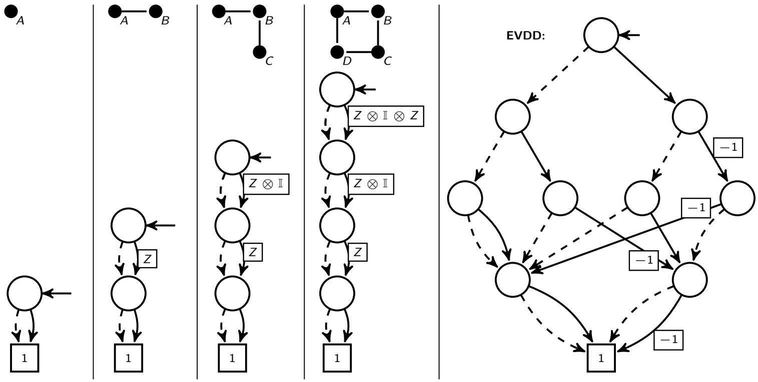
[1] L. Vinkhuijzen, T. Coopmans, D. Elkouss, V. Dunjko, and A. Laarman, "LIMDD: A decision diagram for simulation of quantum computing including stabilizer states," *Quantum*, vol. 7, 2023.

Pauli-LIMDD Examples

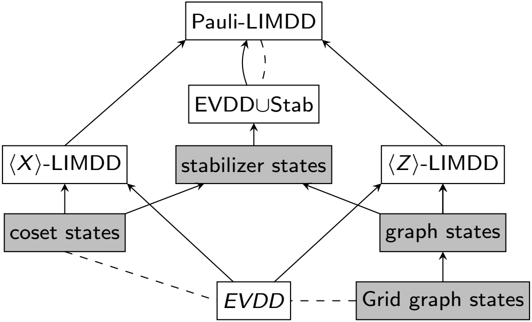


Example $\langle\text{PAULI}\rangle$-Tower LIMDDs for three stabilizer states:
$|e_{GHZ}\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$,
$|e_{+++}\rangle = |+++\rangle$ where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and
$|e_{\varphi}\rangle = \frac{1}{\sqrt{8}}(|000\rangle - |001\rangle + i\,|010\rangle + i\,|011\rangle + i\,|100\rangle + i\,|101\rangle - |110\rangle + |111\rangle)$ with stabilizer group generators $\{\, X \otimes X \otimes X,\ \ -Z \otimes Z \otimes X,\ \ \mathbb{I} \otimes Y \otimes Z \,\}$.

Grid Graph in $\langle Z \rangle$-LIMDD



LIMDD Types

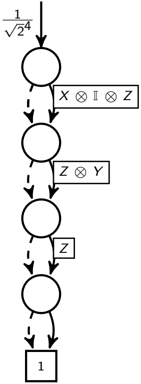


Relations between non-universal classes of quantum states (gray) and decision diagrams, where we consider a diagram as the set of states that it can represent in polynomial size. Solid arrows denote set inclusion. Dashed arrows $D_1 \dashrightarrow D_2$ signify an exponential separation between two classes, i.e., some quantum states have polynomial-size representation in D_1, but only exponential-size in D_2.
By transitivity, *EVDD* is exponentially separated from all representations (not drawn for clarity).

Simulation with LIMDD

Theorem (Stabilizers are small in Pauli-LIMDD [1])

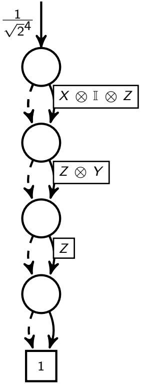
For any stabilizer state, the Pauli-LIMDD is a tower.



Simulation with LIMDD

Theorem (Stabilizers are small in Pauli-LIMDD [1])

For any stabilizer state, the Pauli-LIMDD is a tower.

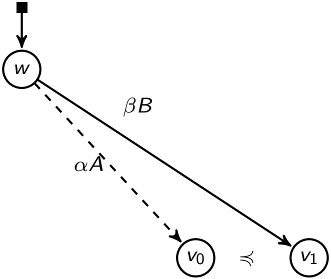


Theorem (Graph-states are small in $\langle Z \rangle$-LIMDD [1])

For any stabilizer state, the $\langle Z \rangle$-LIMDD is a tower.

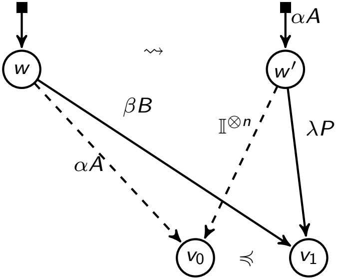
[1] L. Vinkhuijzen, T. Coopmans, D. Elkouss, V. Dunjko, and A. Laarman, "LIMDD: A decision diagram for simulation of quantum computing including stabilizer states." *Quantum*, vol. 7, 2023.

LIMDD Canonicity



$\alpha, \beta \in \mathbb{C}$
$A, B \in \{ I, X, Y, Z \}^{\otimes n}$

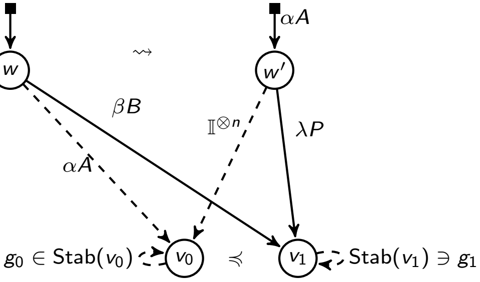
LIMDD Canonicity



$\alpha, \beta \in \mathbb{C}$
$A, B \in \{ I, X, Y, Z \}^{\otimes n}$

$\lambda := \frac{\beta}{\alpha}$
$P := \frac{B}{A}$

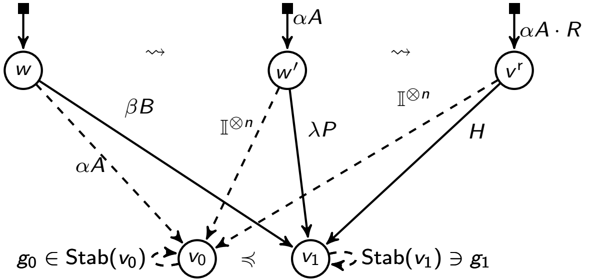
LIMDD Canonicity



$\alpha, \beta \in \mathbb{C}$
$A, B \in \{ I, X, Y, Z \}^{\otimes n}$

$\lambda := \frac{\beta}{\alpha}$
$P := \frac{B}{A}$

LIMDD Canonicity

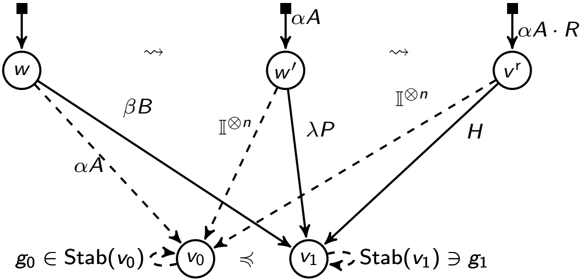


$\alpha, \beta \in \mathbb{C}$
$A, B \in \{ I, X, Y, Z \}^{\otimes n}$

$\lambda := \frac{\beta}{\alpha}$
$P := \frac{B}{A}$

$R := (\lambda X \otimes P)^x \cdot \left(Z^s \otimes (g_0)^{-1} \right)$
$H := (-1)^s \lambda^{(-1)^x} g_0 P g_1$

LIMDD Canonicity



$\alpha, \beta \in \mathbb{C}$
$A, B \in \{ I, X, Y, Z \}^{\otimes n}$

$\lambda := \frac{\beta}{\alpha}$
$P := \frac{B}{A}$

$R := (\lambda X \otimes P)^x \cdot \left(Z^s \otimes (g_0)^{-1} \right)$
$H := (-1)^s \lambda^{(-1)^x} g_0 P g_1$

(This is what causes the $O(n^3)$ overhead, because we compute stabilizer generators for each node and do Gaussian elimination on them.)

Simulation with Pauli-LIMDD

Corollary (1.1 [1])

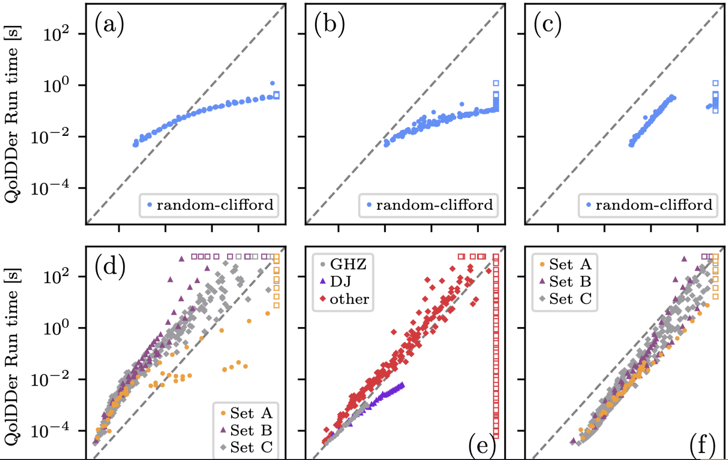
Exact simulation of a Clifford+T circuit with LIMDD requires at most exponential time in the number of T gates and polynomial time in the number of Clifford gates and the number of qubits.
For EVDD, it is exponential in the minimum of H gates and the number of CZ and T gates, and polynomial in the number of other single-qubit Clifford gates and the number of qubits.

[1] A.-J. Quist, T. Coopmans, and A. Laarman, "Exact quantum decision diagrams with scaling guarantees for clifford+ T circuits and beyond," *arXiv preprint arXiv:2602.17775*, 2026.

LIMDD Implementations

Three DD packages implement LIMDD

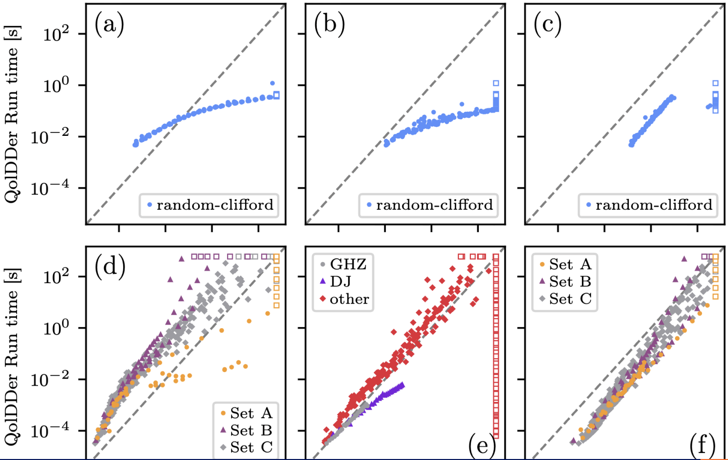
- Pauli-LIMDD in DDSIM [1] in collaboration with Robert Wille (TUM).



LIMDD Implementations

Three DD packages implement LIMDD

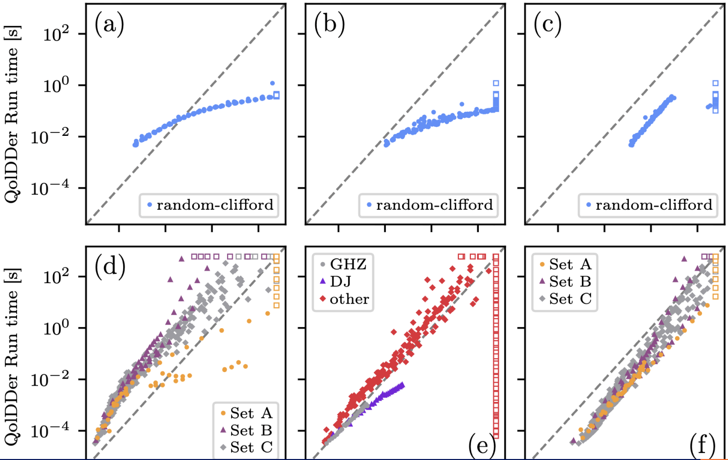
- Pauli-LIMDD in DDSIM [1] in collaboration with Robert Wille (TUM).
- XP-LIMDD (LIMTDD) [2] by Minsheng Ying's group.



LIMDD Implementations

Three DD packages implement LIMDD

- Pauli-LIMDD in DDSIM [1] in collaboration with Robert Wille (TUM).
- XP-LIMDD (LIMTDD) [2] by Minsheng Ying's group.
- Pauli-LIMDD by Tim Coopmans [3] (TU Delft / QuTech)



Classical Representations of Quantum States

From Physics:

- Stabilizer formalism [1] ✓entanglement / ✗structure
- Tensor networks [2] ✗entanglement / ✓structure

From Computer Science:

- Decision Diagrams [4, 5] ✗entanglement / ✓structure
- LIMDD [6] ✓entanglement / ✓structure

[1] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits," *Physical Review A*, vol. 70, 2004.

[2] C. J. Wood, J. D. Biamonte, and D. G. Cory, "Tensor networks and graphical calculus for open quantum systems," *Quantum Info. Comput.*, vol. 15, 2015.

[3] S. Masot-Llima and A. Garcia-Saez, "Stabilizer tensor networks: Universal quantum simulator on a basis of stabilizer states," *Physical Review Letters*, vol. 133, 2024.

[4] G. Viamontes, I. Markov, and J. Hayes, "High-performance QuIDD-based simulation of quantum circuits," in *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, vol. 2, 2004.

[5] D. M. Miller and M. A. Thornton, "QMDD: A decision diagram structure for reversible and quantum circuits," *ISMVL*, 2006.

[6] L. Vinkhuijzen, T. Coopmans, D. Elkouss, V. Dunjko, and A. Laarman, "LIMDD: A decision diagram for simulation of quantum computing including stabilizer states," *Quantum*, vol. 7, 2023.

Classical Representations of Quantum States

From Physics:

- Stabilizer formalism [1] ✓entanglement / ✗structure
- Tensor networks [2] ✗entanglement / ✓structure
- Recently, Stabilizer Tensor Networks [3]. ✓entanglement / ✓structure

From Computer Science:

- Decision Diagrams [4, 5] ✗entanglement / ✓structure
- LIMDD [6] ✓entanglement / ✓structure

[1] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits," *Physical Review A*, vol. 70, 2004.

[2] C. J. Wood, J. D. Biamonte, and D. G. Cory, "Tensor networks and graphical calculus for open quantum systems," *Quantum Info. Comput.*, vol. 15, 2015.

[3] S. Masot-Llima and A. Garcia-Saez, "Stabilizer tensor networks: Universal quantum simulator on a basis of stabilizer states," *Physical Review Letters*, vol. 133, 2024.

[4] G. Viamontes, I. Markov, and J. Hayes, "High-performance QuIDD-based simulation of quantum circuits," in *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, vol. 2, 2004.

[5] D. M. Miller and M. A. Thornton, "QMDD: A decision diagram structure for reversible and quantum circuits," *ISMVL*, 2006.

[6] L. Vinkhuijzen, T. Coopmans, D. Elkouss, V. Dunjko, and A. Laarman, "LIMDD: A decision diagram for simulation of quantum computing including stabilizer states," *Quantum*, vol. 7, 2023.

Classical Representations of Quantum States

From Physics:

- Stabilizer formalism [1] ✓entanglement / ✗structure
- Tensor networks [2] ✗entanglement / ✓structure
- Recently, Stabilizer Tensor Networks [3]. ✓entanglement / ✓structure

From Computer Science:

- Decision Diagrams [4, 5] ✗entanglement / ✓structure
- LIMDD [6] ✓entanglement / ✓structure

But how do these formalisms relate exactly?

[1] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits," *Physical Review A*, vol. 70, 2004.

[2] C. J. Wood, J. D. Biamonte, and D. G. Cory, "Tensor networks and graphical calculus for open quantum systems," *Quantum Info. Comput.*, vol. 15, 2015.

[3] S. Masot-Llima and A. Garcia-Saez, "Stabilizer tensor networks: Universal quantum simulator on a basis of stabilizer states," *Physical Review Letters*, vol. 133, 2024.

[4] G. Viamontes, I. Markov, and J. Hayes, "High-performance QuIDD-based simulation of quantum circuits," in *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, vol. 2, 2004.

[5] D. M. Miller and M. A. Thornton, "QMDD: A decision diagram structure for reversible and quantum circuits," *ISMVL*, 2006.

[6] L. Vinkhuijzen, T. Coopmans, D. Elkouss, V. Dunjko, and A. Laarman, "LIMDD: A decision diagram for simulation of quantum computing including stabilizer states," *Quantum*, vol. 7, 2023.

Overview

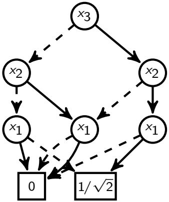
- **Background**
- **Knowledge Representation for QI**
  - BDDs for Quantum States
  - LIMDD Definition
  - Quantum Knowledge Compilation Map
- **Conclusion**

Decision Diagrams, RBM, and MPS

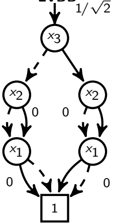
Vector

$$\begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

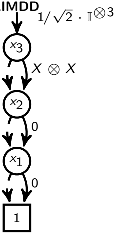
MTDD



EVDD

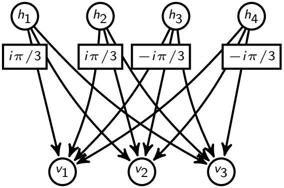


LIMDD



RBM

hidden layer



visible layer

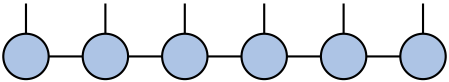
Tensor Train

$A_3^0 = \begin{bmatrix} 1 & 0 \end{bmatrix}$ $\quad$ $A_3^1 = \begin{bmatrix} 0 & 1 \end{bmatrix}$

$A_2^0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ $\quad$ $A_2^1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

$A_1^0 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}$ $\quad$ $A_1^1 = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}$

Matrix Product States (Tensor Trains)



Definition (MPS on n qubits)

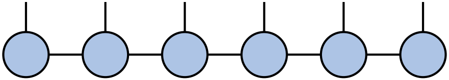
An MPS M is a series of $2n$ matrices of the right dimensions.

$A_1^0, \quad A_2^0, \quad \ldots, \quad A_n^0$

$A_1^1, \quad A_2^1, \quad \ldots, \quad A_n^1$

The interpretation $|M\rangle$ is determined as $\langle \vec{x}|M\rangle = A_n^{x_n} \cdot \cdots \cdot A_2^{x_2} \cdot A_1^{x_1}$ for $\vec{x} \in \{0,1\}^n$.

Matrix Product States (Tensor Trains)



Definition (MPS on n qubits)

An MPS M is a series of $2n$ matrices of the right dimensions.

$A_1^0, \quad A_2^0, \quad \ldots, \quad A_n^0$

$A_1^1, \quad A_2^1, \quad \ldots, \quad A_n^1$

The interpretation $|M\rangle$ is determined as $\langle \vec{x}|M\rangle = A_n^{x_n} \cdot \cdots \cdot A_2^{x_2} \cdot A_1^{x_1}$ for $\vec{x} \in \{0,1\}^n$.

MPS

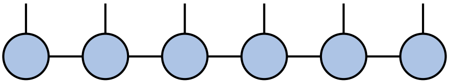
$$|A\rangle = \begin{array}{cl} \left[\begin{array}{c} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{array}\right] & \begin{array}{l} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{array} \end{array}$$

$A_3^0 = \begin{bmatrix} 1 & 0 \end{bmatrix}$ $\quad A_3^1 = \begin{bmatrix} 0 & 1 \end{bmatrix}$

$A_2^0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ $\quad A_2^1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$

$A_1^0 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}$ $\quad A_1^1 = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}$

Matrix Product States (Tensor Trains)



Definition (MPS on n qubits)

An MPS M is a series of $2n$ matrices of the right dimensions.

$A_1^0, \quad A_2^0, \quad \dots, \quad A_n^0$

$A_1^1, \quad A_2^1, \quad \dots, \quad A_n^1$

The interpretation $|M\rangle$ is determined as $\langle \vec{x} | M \rangle = A_n^{x_n} \cdot \dots \cdot A_2^{x_2} \cdot A_1^{x_1}$ for $\vec{x} \in \{0,1\}^n$.

MPS

$$|A\rangle = \begin{array}{c} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix} \end{array} \begin{array}{l} 000 \\ 001 \\ 010 \\ 011 \\ 100 \\ 101 \\ 110 \\ 111 \end{array}$$

$$A_3^0 = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad A_3^1 = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$A_2^0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \qquad A_2^1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$A_1^0 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} \qquad A_1^1 = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$

$$\langle 111 | A \rangle = A_1^1 \cdot A_2^1 * A_3^1$$

EVDD vs MPS

Theorem

MPS is at least as succinct as EVDD.

EVDD vs MPS

Theorem

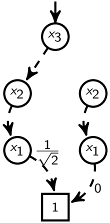
MPS is at least as succinct as EVDD.

Proof:

$$A_3^0 = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$A_2^0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$A_1^0 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix}$$



EVDD vs MPS

Theorem

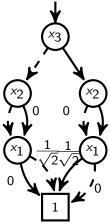
MPS is at least as succinct as EVDD.

Proof:

$$A_3^0 = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad A_3^1 = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$A_2^0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \qquad A_2^1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$A_1^0 = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ 0 \end{bmatrix} \qquad A_1^1 = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}$$



EVDD vs MPS

Theorem

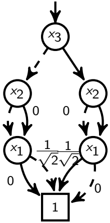
MPS is at least as succinct as EVDD.

Proof:

$$A_3^0 = \begin{bmatrix}1 & 0\end{bmatrix} \qquad A_3^1 = \begin{bmatrix}0 & 1\end{bmatrix}$$

$$A_2^0 = \begin{bmatrix}1 & 0 \\ 0 & 0\end{bmatrix} \qquad A_2^1 = \begin{bmatrix}0 & 0 \\ 0 & 1\end{bmatrix}$$

$$A_1^0 = \begin{bmatrix}\frac{1}{\sqrt{2}} \\ 0\end{bmatrix} \qquad A_1^1 = \begin{bmatrix}0 \\ \frac{1}{\sqrt{2}}\end{bmatrix}$$



Theorem

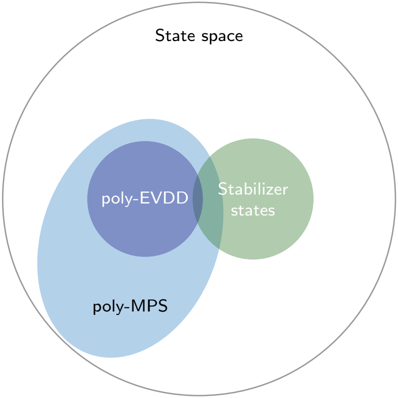
There is a family of states with polynomial-size MPS but exponential-size EVDD.

$$|\varphi_n\rangle = \sum_{x \in \{0,1\}^n} (x)_2 \, |x\rangle$$

$$A_n^0 = \begin{bmatrix}1 & 0\end{bmatrix} \qquad A_j^0 = \begin{bmatrix}1 & 0 \\ 0 & 1\end{bmatrix} \qquad A_1^0 = \begin{bmatrix}0 \\ 1\end{bmatrix} \tag{1}$$

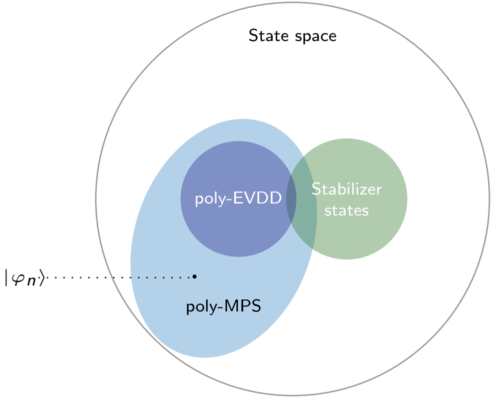
$$A_n^1 = \begin{bmatrix}1 & 2^{n-1}\end{bmatrix} \qquad A_j^1 = \begin{bmatrix}1 & 2^{j-1} \\ 0 & 1\end{bmatrix} \qquad A_1^1 = \begin{bmatrix}1 \\ 1\end{bmatrix} \tag{2}$$

Succinctness Separations



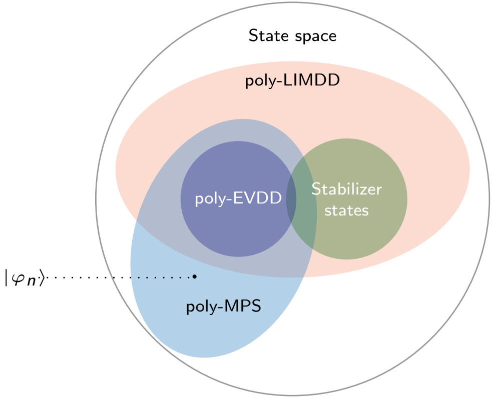
[1] L. Vinkhuijzen, T. Coopmans, and A. Laarman, "A knowledge compilation map for quantum information," in *AAAI*, vol. 28, 2026.

Succinctness Separations



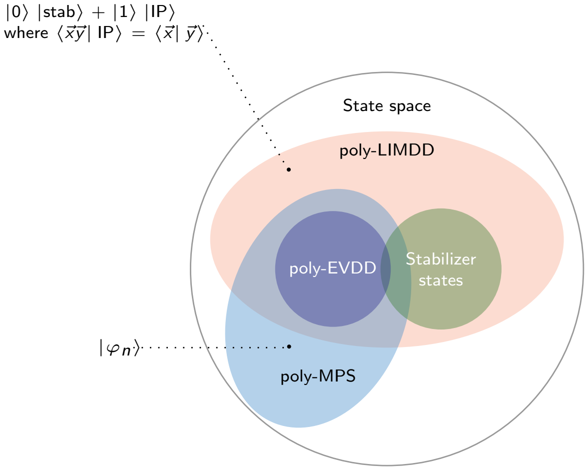
[1] L. Vinkhuijzen, T. Coopmans, and A. Laarman, "A knowledge compilation map for quantum information," in *AAAI*, vol. 28, 2026.

Succinctness Separations



[1] L. Vinkhuijzen, T. Coopmans, and A. Laarman, "A knowledge compilation map for quantum information," in *AAAI*, vol. 28, 2026.

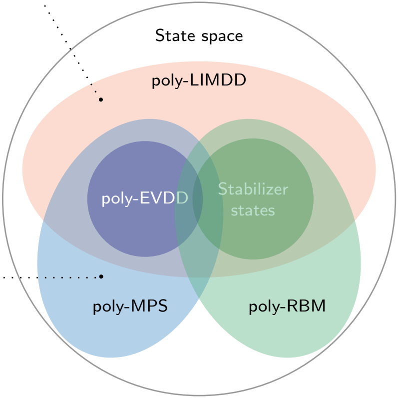
Succinctness Separations



[1] L. Vinkhuijzen, T. Coopmans, and A. Laarman, "A knowledge compilation map for quantum information," in *AAAI*, vol. 28, 2026.

Succinctness Separations

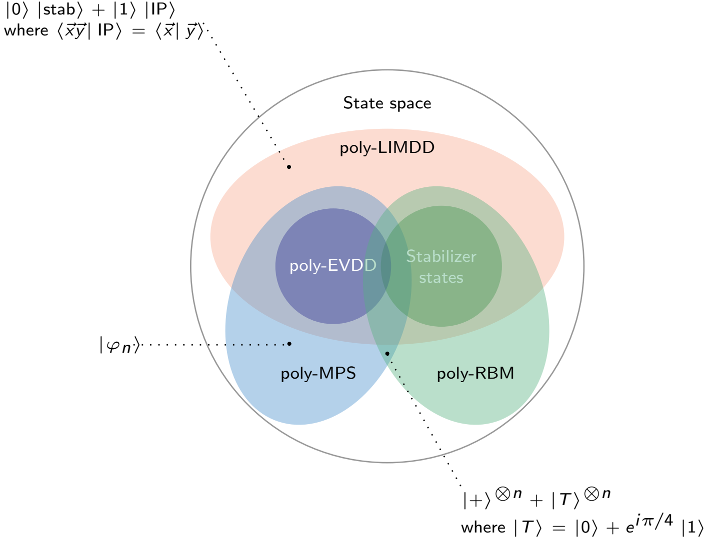
$|0\rangle\,|\text{stab}\rangle + |1\rangle\,|\text{IP}\rangle$
where $\langle \vec{x}\vec{y}|\,\text{IP}\rangle = \langle \vec{x}|\,\vec{y}\rangle$



$|\varphi_n\rangle$

[1] L. Vinkhuijzen, T. Coopmans, and A. Laarman, "A knowledge compilation map for quantum information," in *AAAI*, vol. 28, 2026.

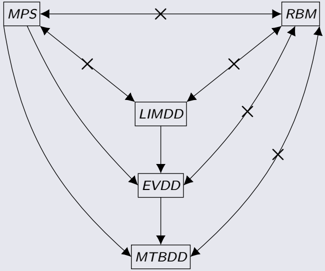
Succinctness Separations



[1] L. Vinkhuijzen, T. Coopmans, and A. Laarman, "A knowledge compilation map for quantum information," in *AAAI*, vol. 28, 2026.

A Knowledge Compilation Map for Quantum Information

Theorem (Succinctness and tractability of DD, MPS, and RBM [1])

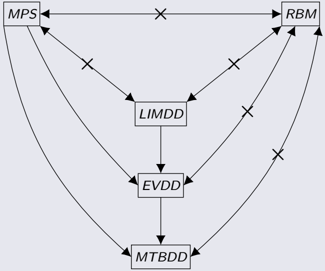


Succinctness separations.

[1] L. Vinkhuijzen, T. Coopmans, and A. Laarman, "A knowledge compilation map for quantum information," in *AAAI*, vol. 28, 2026.

A Knowledge Compilation Map for Quantum Information

Theorem (Succinctness and tractability of DD, MPS, and RBM [1])



Succinctness separations.

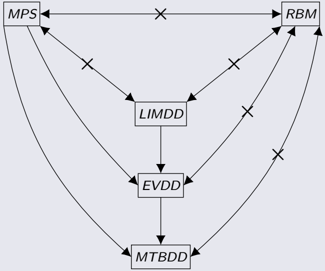
	Queries					Manipulation operations						
	Sample	**Measure**	**Equal**	**InnerProd**	**Fidelity**	**Addition**	**Hadamard**	**X,Y,Z**	**CZ**	**Swap**	**Local**	**T**-*gate*
Vector	✓'	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MTBDD	✓'	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EVDD	✓'	✓	✓	✓	✓	✖	✖	✓	✓	✖	✖	✓
LIMDD	✓'	✓	✓	○	○	✖	✖	✓	✓	✖	✖	✓
MPS	✓'	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
RBM	✓'	?	?	○	○	?	?	✓	✓	✓	?	✓

✓ (✖) means the operation is (not) supported in polytime.
○ means the operation is not supported unless $P = NP$.

[1] L. Vinkhuijzen, T. Coopmans, and A. Laarman, “A knowledge compilation map for quantum information,” in *AAAI*, vol. 28, 2026.

A Knowledge Compilation Map for Quantum Information

Theorem (Succinctness and tractability of DD, MPS, and RBM [1])



Succinctness separations.

	Queries					Manipulation operations						
	Sample	**Measure**	**Equal**	**InnerProd**	**Fidelity**	**Addition**	**Hadamard**	**X,Y,Z**	**CZ**	**Swap**	**Local**	**T**-*gate*
Vector	✓'	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
MTBDD	✓'	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EVDD	✓'	✓	✓	✓	✓	✖	✖	✓	✓	✖	✖	✓
LIMDD	✓'	✓	✓	○	○	✖	✖	✓	✓	✖	✖	✓
MPS	✓'	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
RBM	✓'	?	?	○	○	?	?	✓	✓	✓	?	✓

✓ (✖) means the operation is (not) supported in polytime.
○ means the operation is not supported unless $P = NP$.

[1] L. Vinkhuijzen, T. Coopmans, and A. Laarman, "A knowledge compilation map for quantum information," in *AAAI*, vol. 28, 2026.

Relating Circuits and Tensor Networks

Tensor network type		Circuit type
Tensor Train (TT, MPS)	$\Leftrightarrow$	Non-Deterministic EVDD
Deterministic TT	$\Leftrightarrow$	EVDD
Tree Tensor Network (TTN)	$\Leftrightarrow$	EV-SDNNF
Decision TTN	$\Leftrightarrow$	Decision EV-SDNNF
Deterministic TTN	$\Leftrightarrow$	Deterministic EV-SDNNF

Table: Tensor network and decision diagram equivalences.

[1] Quist, F. Bartra, de Colnet, van de Wetering, and Laarman, "From tensor networks to tractable circuits, and back," *KR*, 2026.

Fidelity is Hard for RBM and LIMDD

Theorem

RBM and LIMDD are not tractable for **Fidelity**, *unless ETH fails.*

Fidelity is Hard for RBM and LIMDD

Theorem

RBM and LIMDD are not tractable for **Fidelity**, *unless ETH fails.*

Proof:

Let $|D_n^k\rangle$ be the Dicke state on n qubits.

Fidelity is Hard for RBM and LIMDD

Theorem

RBM and LIMDD are not tractable for **Fidelity**, *unless ETH fails.*

Proof:

Let $|D_n^k\rangle$ be the Dicke state on n qubits.

Let G_n be an n-vertex graph and $|G_n\rangle$ be the corresponding graph state.

Fidelity is Hard for RBM and LIMDD

Theorem

RBM and LIMDD are not tractable for **Fidelity**, *unless ETH fails.*

Proof:

Let $|D_n^k\rangle$ be the Dicke state on n qubits.

Let G_n be an n-vertex graph and $|G_n\rangle$ be the corresponding graph state.

The #EVEN SUBGRAPHS problem reduces to computing the fidelity $|\langle D_n^k|G_n\rangle|^2$

Definition (#EVEN SUBGRAPHS)

Input: A graph $G = (V, E)$, an integer k
Output: The number of induced k-vertex subgraphs with an even number of edges

Fidelity is Hard for RBM and LIMDD

Theorem

RBM and LIMDD are not tractable for **Fidelity**, *unless ETH fails.*

Proof:

Let $|D_n^k\rangle$ be the Dicke state on n qubits.

Let G_n be an n-vertex graph and $|G_n\rangle$ be the corresponding graph state.

The #EVEN SUBGRAPHS problem reduces to computing the fidelity $|\langle D_n^k|G_n\rangle|^2$

Definition (#EVEN SUBGRAPHS)

Input: A graph $G = (V, E)$, an integer k
Output: The number of induced k-vertex subgraphs with an even number of edges

Theorem (Jerrum and Meeks)

If #EVEN SUBGRAPHS is not in P, *unless ETH is false.*

Actually, omitting detail: #EVEN-ODD-SUBGRAPH-DIFFERENCE

[1] M. Jerrum and K. Meeks, "The parameterised complexity of counting even and odd induced subgraphs," *Combinatorica*, vol. 37, 2017.

Open Questions

To do

- Can we have LIM-TN?
- Can we get rid of cubic overhead in LIMDD operations?
- Can we use LIMDDs to get new asymptotic results?

Overview

- **Background**
- **Knowledge Representation for QI**
- **Conclusion**

Open directions — for your PhD

- Which succinctness separations among LIMDD/ QMDD / MPS / RBM / stabilizer are still open?

Open directions — for your PhD

- Which succinctness separations among LIMDD/ QMDD / MPS / RBM / stabilizer are still open?
- Tractable query sets: complete the map — what can each representation answer in poly time?

Open directions — for your PhD

- Which succinctness separations among LIMDD/ QMDD / MPS / RBM / stabilizer are still open?
- Tractable query sets: complete the map — what can each representation answer in poly time?
- Canonical forms & minimization for LIMDD.

Open directions — for your PhD

- Which succinctness separations among LIMDD/ QMDD / MPS / RBM / stabilizer are still open?
- Tractable query sets: complete the map — what can each representation answer in poly time?
- Canonical forms & minimization for LIMDD.
- Sharper hardness for fidelity / inner product.

Open directions — for your PhD

- Which succinctness separations among LIMDD/ QMDD / MPS / RBM / stabilizer are still open?
- Tractable query sets: complete the map — what can each representation answer in poly time?
- Canonical forms & minimization for LIMDD.
- Sharper hardness for fidelity / inner product.
- A full knowledge-compilation map for quantum: fill the empty cells.

Open directions — for your PhD

- Which succinctness separations among LIMDD/ QMDD / MPS / RBM / stabilizer are still open?
- Tractable query sets: complete the map — what can each representation answer in poly time?
- Canonical forms & minimization for LIMDD.
- Sharper hardness for fidelity / inner product.
- A full knowledge-compilation map for quantum: fill the empty cells.

Clean, theory-friendly problems — come collaborate.

Takeaways and Thank Yous

- LIMDD offers the first tight unification of the stabilizer formalism with a DD/TN-like data structure.
- LIMDD implementations are not there yet.

Takeaways and Thank Yous

- LIMDD offers the first tight unification of the stabilizer formalism with a DD/TN-like data structure.
- LIMDD implementations are not there yet.

Takeaways and Thank Yous

- LIMDD offers the first tight unification of the stabilizer formalism with a DD/TN-like data structure.
- LIMDD implementations are not there yet.







Dimitrios Thanos, Alejandro Villoria, Sebastiaan Brand, Arend-Jan Quist, Jingyi Mei, Lieuwe Vinkhuijzen, Tim Coopmans

https://github.com/cda-tum/mqt-limd

[1] L. Vinkhuijzen, T. Coopmans, D. Elkouss, V. Dunjko, and A. Laarman, "LIMDD: A decision diagram for simulation of quantum computing including stabilizer states," *Quantum*, vol. 7, 2023.

[2] L. Vinkhuijzen, T. Coopmans, and A. Laarman, "A knowledge compilation map for quantum information," in *AAAI*, vol. 28, 2026.

[3] J. Mei, M. Bonsangue, and A. Laarman, "Simulating quantum circuits by model counting," in *CAV*, 2024.

[4] J. Mei, T. Coopmans, M. Bonsangue, and A. Laarman, "Equivalence checking of quantum circuits by model counting," in *IJCAR*, 2024.

[5] D. Thanos, A. Villoria, S. Brand, A.-J. Quist, J. Mei, et al., "Automated reasoning in quantum circuit compilation," in *Model Checking Software (SPIN)*, 2024.

[6] D. Thanos, T. Coopmans, and A. Laarman, "Fast equivalence checking of quantum circuits of Clifford gates," in *ATVA*, 2023.