Toward Entailment Checking: Explore Eigenmarking Search

Tatpong Katanyukul
Computer Engineering, Khon Kaen University, Thailand

inCACCT 2025, 17-18 April, Chandigarh University, India

Big Picture

- Entailment is central to logic reasoning.
- Model checking goes through all combinations of logical symbols for validation of entailment: $O(2^n)$.
- Our work is to propose improved quantum search targeting a more efficient model checking.

Logic Entailment

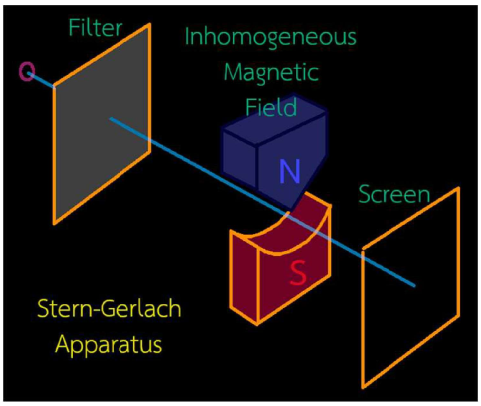
$KB \vDash \alpha$ if and only if, in every model (truth scenario) in which KB is true, α is true.

KB:
- Durians are spiky.
- Durians are yummy.

α_1: Montong durian is spiky.	$KB \vDash \alpha_1$
α_2: Montong durian is not spiky.	$KB \nvDash \alpha_2$
α_3: There is life on Mars.	$KB \nvDash \alpha_3$
α_4: There is no life on Mars.	$KB \nvDash \alpha_4$

Note
- $KB \vDash \alpha$ means α agrees with what KB said.
- $KB \nvDash \alpha$ means α is not what is said by KB.

Quantum Computing and Quantum Mechanical Properties



- Quantum computing utilizes quantum mechanical properties for computing.
- The quantum effect is more prominent at a small scale.
  - Linear evolution
  - Measurement
  - **Superposition**, Entanglement, Tunneling.

Quantum State and Superposition

- Superposition: a quantum state is a combination of eigenstates.

$$|\psi\rangle = c_1|\psi_1\rangle + c_2|\psi_2\rangle + c_3|\psi_3\rangle + \cdots + c_N|\psi_N\rangle$$

or vector representation

$$|\psi\rangle = [c_1 \quad c_2 \quad c_3 \quad \cdots \quad c_N]^T$$

Quantum Computing

In short, we can control quantum state evolution by unitary operator *U* through manipulation of the system energy,

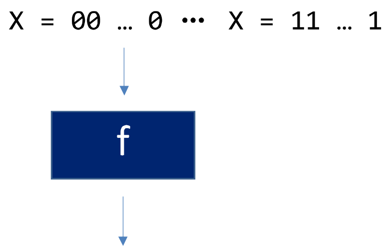
$$|\psi'\rangle = U\,|\psi\rangle.$$

And we can measure the state and collapse it to one of the eigenstates with probability,

$$\Pr[\,|\psi'\rangle = |\psi_i\rangle\,] = |c_i|^2.$$

Grover Search

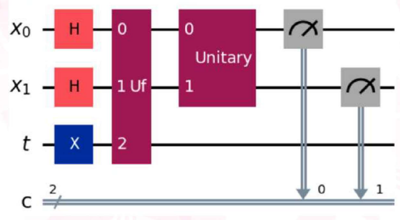
Classical



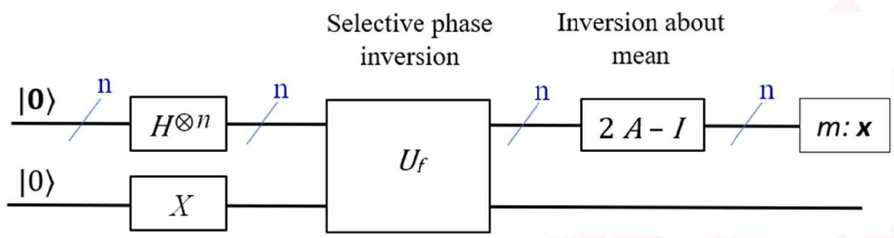
- Problem: find an answer $x' \in \{0,1\}^n$: $f(x') = 1$
- Promise: only one x': $f(x') = 1$
  and $f(x) = 0$ for all $x \neq x'$.

- Classical approach: trial-and-error
  - Average computation cost ~ $O\left(\frac{N}{2}\right) = O(2^{n-1})$
  - All possible candidates $N = 2^n$.

Grover search



Grover Algorithm: Key Ideas



- Evolve the probability amplitude of the answer eigenstate such that when measured, the answer is more likely to be observed.

- Implementation:
  - Selective phase inversion: mark the answer.
  - Inversion about the mean: amplify the answer's amplitude.

Shortcomings of Original Grover in Entailment Context

- Designed for a lone match search.
  - Mitigation:
    - Probabilistic control over # applications.
    - Time-out to handle a no-winner case.

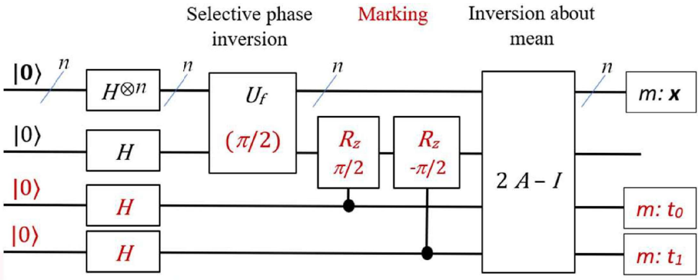
- Entailment checking is likely to have multiple matches or no match at all

  - No match $\Rightarrow$ no violation: the entailment is validated.

Our Approach

- Use additional qubit(s)
  - Additional qubits
    - More effective eigenstates:
      maintain minority condition of Grover search
    - Facilitate easy identification of no-winner case

Eigenmarking

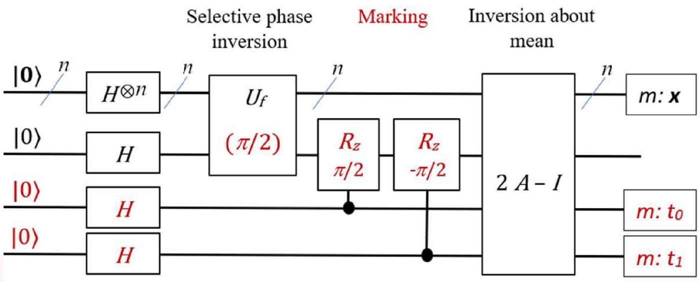


- Tag 00, $\phi(\boldsymbol{x}') = \frac{\pi}{2} + 0 + 0 = \frac{\pi}{2}$ and $\phi(\boldsymbol{x}) = 0 + 0 + 0 = 0$.
- Tag 01, $\phi(\boldsymbol{x}') = \frac{\pi}{2} + 0 + \frac{\pi}{2} = \pi$ and $\phi(\boldsymbol{x}) = 0 + 0 + \frac{\pi}{2} = \frac{\pi}{2}$.
- Tag 10, $\phi(\boldsymbol{x}') = \frac{\pi}{2} - \frac{\pi}{2} + 0 = 0$ and $\phi(\boldsymbol{x}) = 0 - \frac{\pi}{2} + 0 = -\frac{\pi}{2}$.
- Tag 11, $\phi(\boldsymbol{x}') = \frac{\pi}{2} = \frac{\pi}{2}$ and $\phi(\boldsymbol{x}) = 0 = 0$.

- With winners, tag 01 has the answer(s): $\phi(\boldsymbol{x}') = \pi$ while others having $-\frac{\pi}{2}, 0, \frac{\pi}{2}$.
  - But all-winner and no-winner may look the same.

Eigenmarking: Cases

Tag	$\phi(x')$	$\phi(x)$
00	$\pi/2$	0
01	π	$\pi/2$
10	0	$-\pi/2$
11	$\pi/2$	0



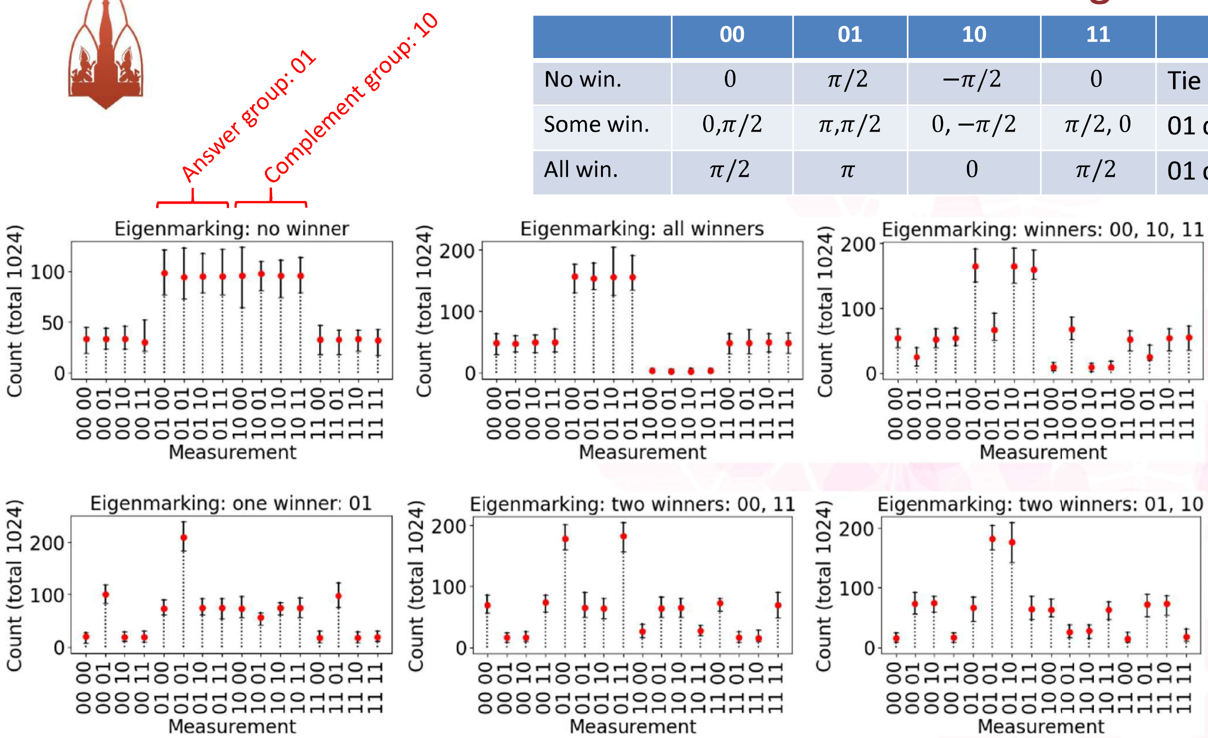
	00	01	10	11
No win.	0	$\pi/2$	$-\pi/2$	0
Some win.	$0, \pi/2$	$\pi, \pi/2$	$0, -\pi/2$	$\pi/2, 0$
All win.	$\pi/2$	π	0	$\pi/2$

Our target

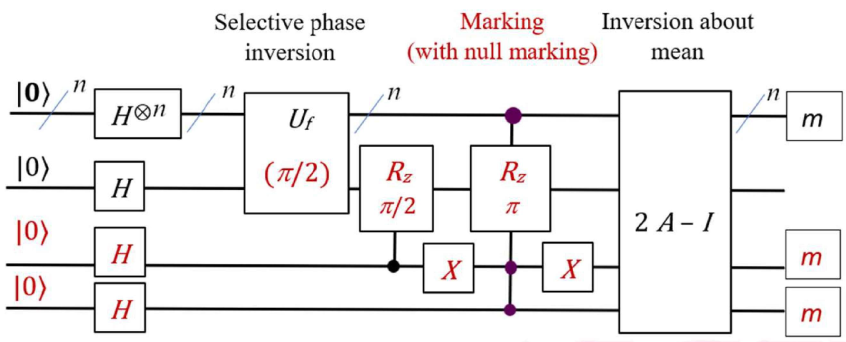
Complementary, esp., in no win.

Results: Eigenmarking

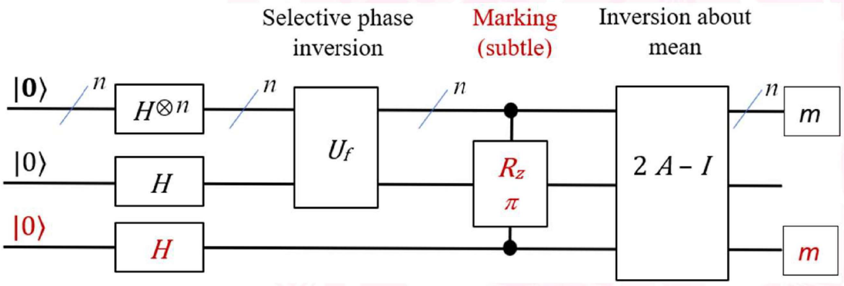
	00	01	10	11	
No win.	0	$\pi/2$	$-\pi/2$	0	Tie 01 and 10
Some win.	$0, \pi/2$	$\pi, \pi/2$	$0, -\pi/2$	$\pi/2, 0$	01 dominates
All win.	$\pi/2$	π	0	$\pi/2$	01 dominates



Null and Subtle Eigenmarkings



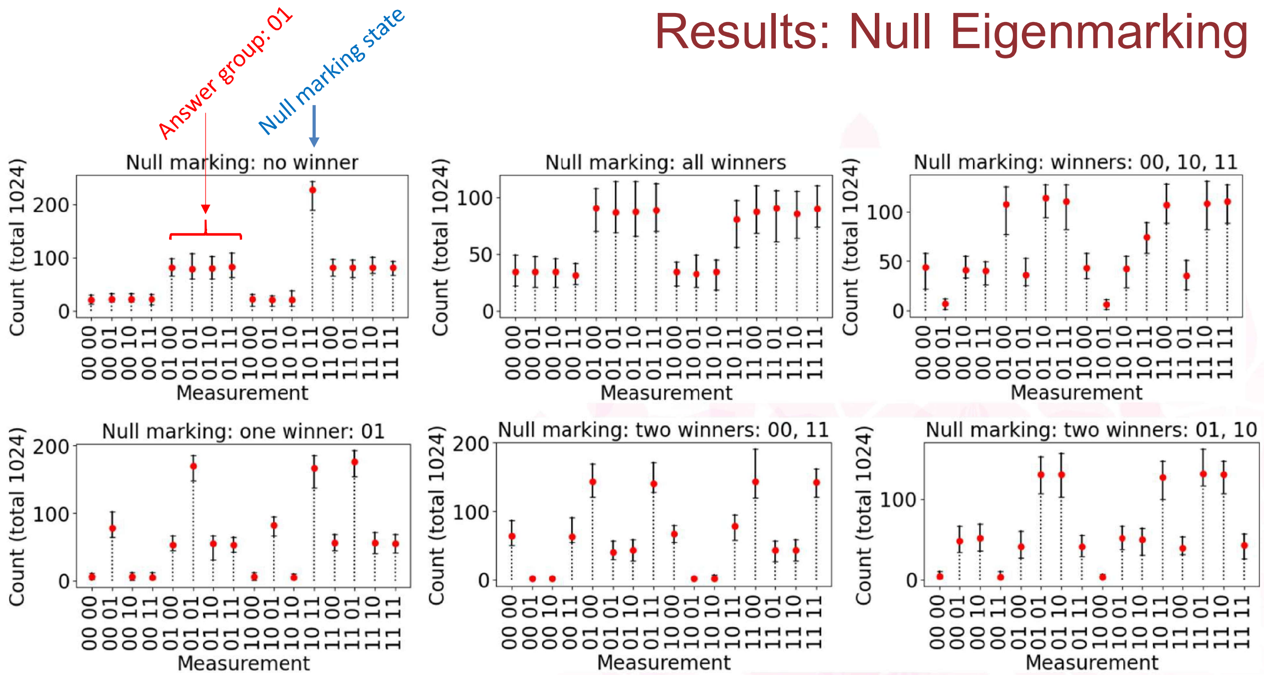
Null marking
- Explicit no-winner state 10 11
- In entailment checking, No-winner ~ no violation!



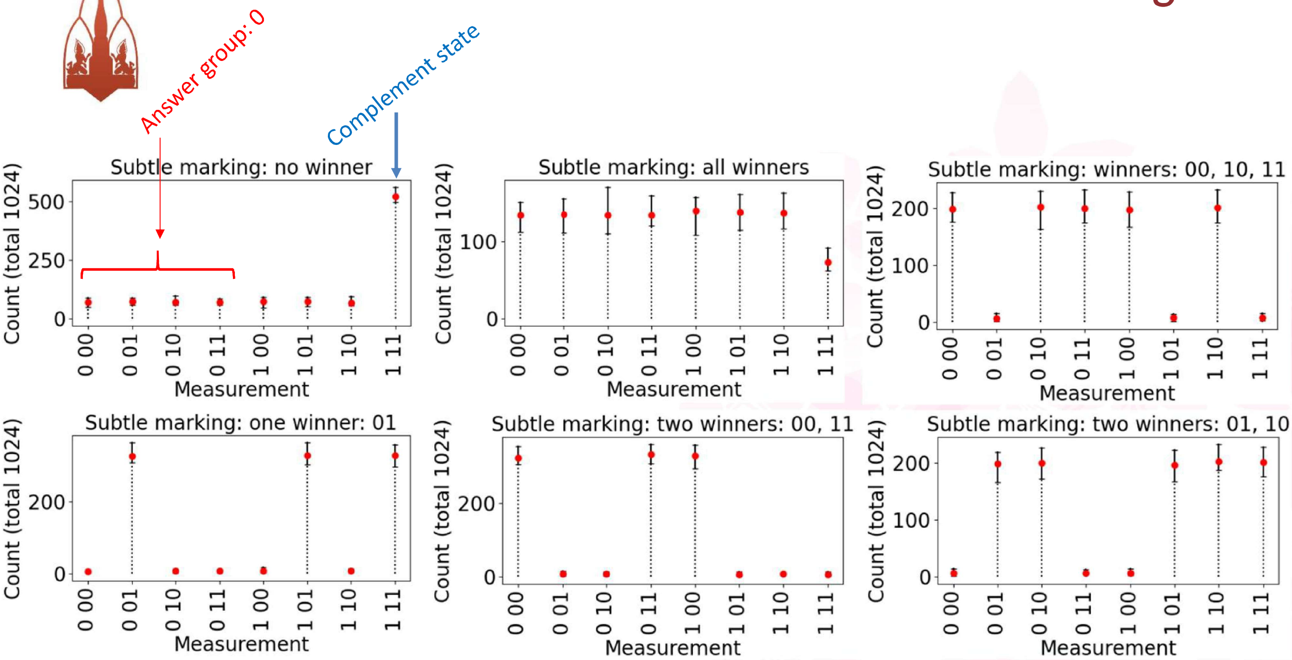
Subtle marking
- Only one extra qubit!
- No change to U_f

Cons
- Multiple-qubit control

Results: Null Eigenmarking



Results: Subtle Eigenmarking



Final Results: Get The Winner

Relative difference between #counts (observed) of winning states and ones of non-winning state.

$$W = (c - c')/c'$$

Scheme	Relative winning margin		Local prefix
	Global	Local	
Eigen.	[0.57, **1.10**, 1.76](0.2)	[0.67, **1.49**, 2.60](0.3)	01
Null.	[-0.44, **-0.09**, 0.27](0.1)	[0.62, **1.82**, 4.74](0.5)	01
Subtle.	[-0.37, **0.31**, 6.12](1.4)	[0.28, **25.72**, 197.00](15.8)	0

E.g., $W = 1.1$ means that chance of seeing the winning state $c \approx 1.1\,c' + c' \approx 2.1\,c'$.

Final Results: Some Win VS No Win

Scheme	Distinguishability			
	Worst-case		Average-case	
	D	$D/\lvert\bar{M}_0\rvert$	d	$d/\lvert\bar{M}_0\rvert$
Eigen.	0.190	19.000	0.532	53.188
Null.	0.220	0.524	0.548	1.306
Subtle.	0.550	0.753	1.140	1.561

Distinguishability ~ gap between at least some win vs no win.

Worst-case: worst score of the winner *vs* best score of the non-winner

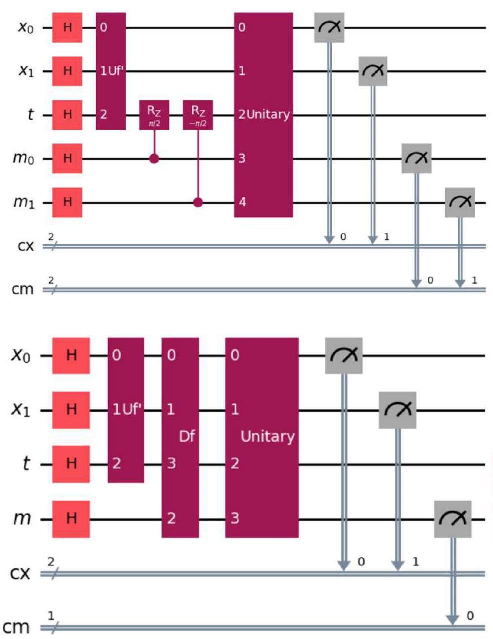
$$D = \frac{\min_{i>0} \min M_i - \max M_0}{|\max M_0|}$$

Conclusion

- The ideas work!   All three schemes work fine.
- Quality of outcomes
  - Eigenmarking
    - Better at suppressing chances of dummy states: best global winning margin.
    - Quite well on distinguishability: best relative scores.
  - Subtle marking
    - Quite well on every aspect:
      - best local winning margin,
      - best absolute distinguishability.

- Architectural aspect: subtle marking requires less, but needs multiple-qubit controls.
- Limitations: *Scalability?* (more qubits) *Reliability?* (theoretical analysis) *Robustness?* (real QC)

Architectures



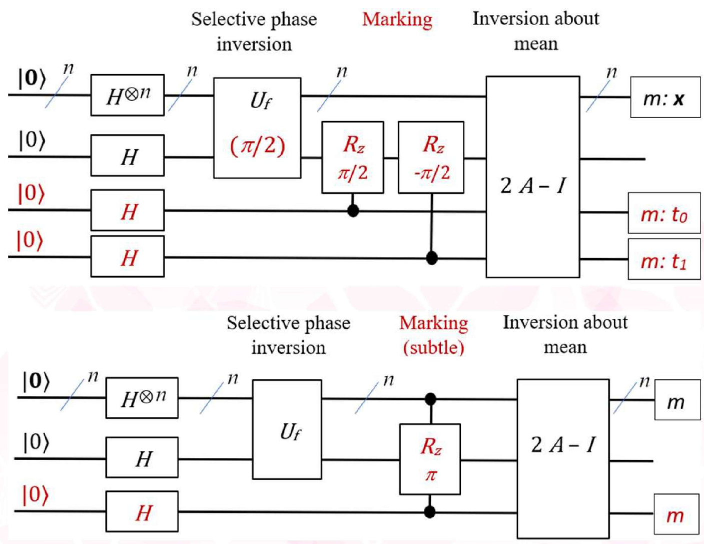


image: https://commons.wikimedia.org/wiki/File:%E0%B8%A7%E0%B8%B1%E0%B8%94%E0%B8%9E%E0%B8%A3%E0%B8%B0%E0%B8%98%E0%B8%B2%E0%B8%95%E0%B8%B8%E0%B8%82%E0%B8%B2%E0%B8%A1%E0%B9%81%E0%B8%81%E0%B9%88%E0%B8%99.jpg

Model Checking

$KB \vDash \alpha$ if and only if, in **every truth scenario** in which KB is true, α is true.

KB:
- Durians are spiky.
- Durians are yummy.

α_1: Montong durian is spiky.
α_2: Montong durian is not spiky.
α_3: There is life on Mars.
α_4: There is no life on Mars.

Spiky Montong	Life on Mars	KB	α_1	α_2	α_3	α_4	
F	F	F	F	T	F	T	
F	T	F	F	T	T	F	$O(2^n)$
T	F	**T**	T	F	F	T	
T	T	**T**	T	F	T	F	
			$KB \vDash \alpha_1$	$KB \nvDash \alpha_2$	$KB \nvDash \alpha_3$	$KB \nvDash \alpha_4$	