Title: Nondeterministic testing of sequential quantum logic propositions on a quantum computer

Date: Jul 19, 2005 04:00 PM

URL: http://pirsa.org/05070102

Abstract:

Nondeterministic testing of SQL Propositions on a QC

Matthew Leifer

Perimeter Institute for Theoretical Physics

QICL Workshop 17th-22nd July 2005

Outline

1) Introduction

Logic, complexity and models of Computing

2) Sequential Quantum Logic

3) Testing an SQL proposition: Example

4) Generalizations

Higher dimensions, multiple propositions

6) Open questions

Nondeterministic testing of SQL Propositions on a QC

Matthew Leifer

Perimeter Institute for Theoretical Physics

QICL Workshop 17th-22nd July 2005

Outline

1) Introduction

Logic, complexity and models of Computing

2) Sequential Quantum Logic

3) Testing an SQL proposition: Example

4) Generalizations

Higher dimensions, multiple propositions

6) Open questions

1) Introduction

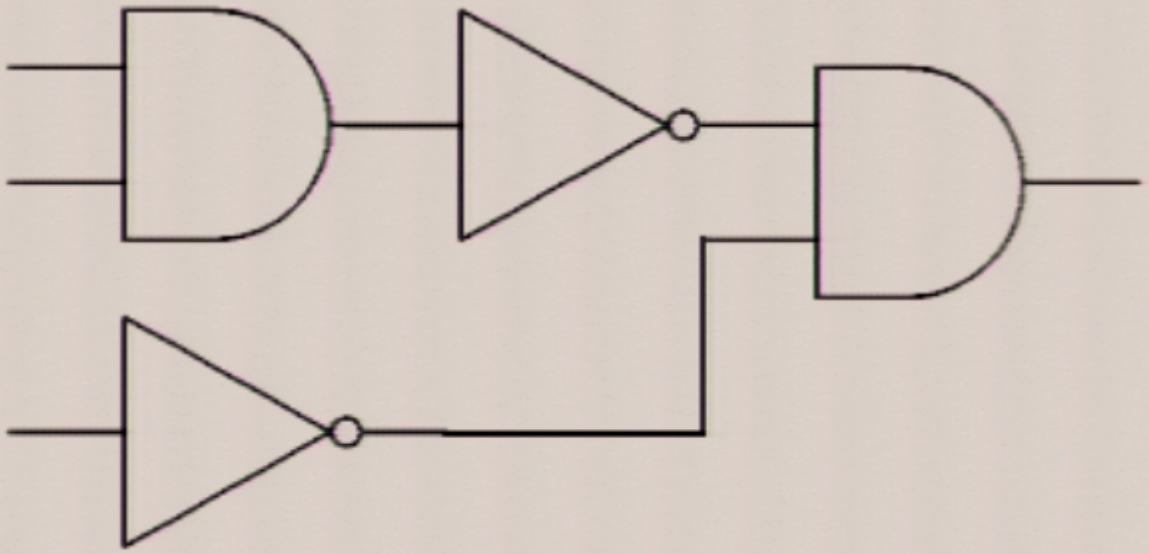
Classical Computational Complexity

Cook-Levin: SATISFIABILITY is NP complete.

Given a Boolean formula, decide if there are truth value assignments to the elementary propositions that make the formula true.

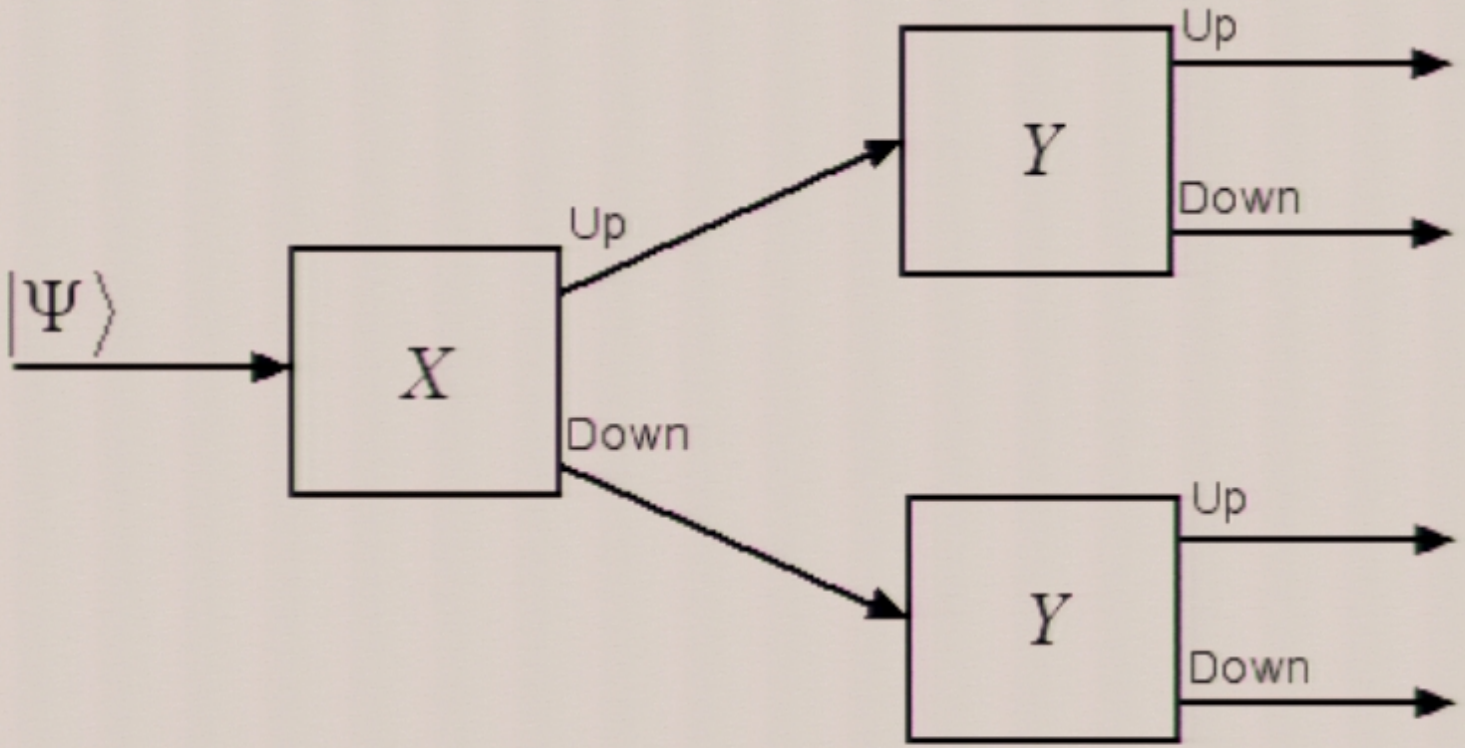
1) Introduction

There is a model of classical computing based directly on Boolean logic.



2) Sequential Quantum Logic

- Developed in late 70's/early 80's by Stachow and Mittlestaedt.
- Extended by Isham et. al. in mid 90's as a possible route to QGravity.



2) SQL: Structure

$$\langle L, S, \sqcap, \neg, (,) \rangle$$

L is a set of elementary propositions and S is the set of sequential propositions given by:

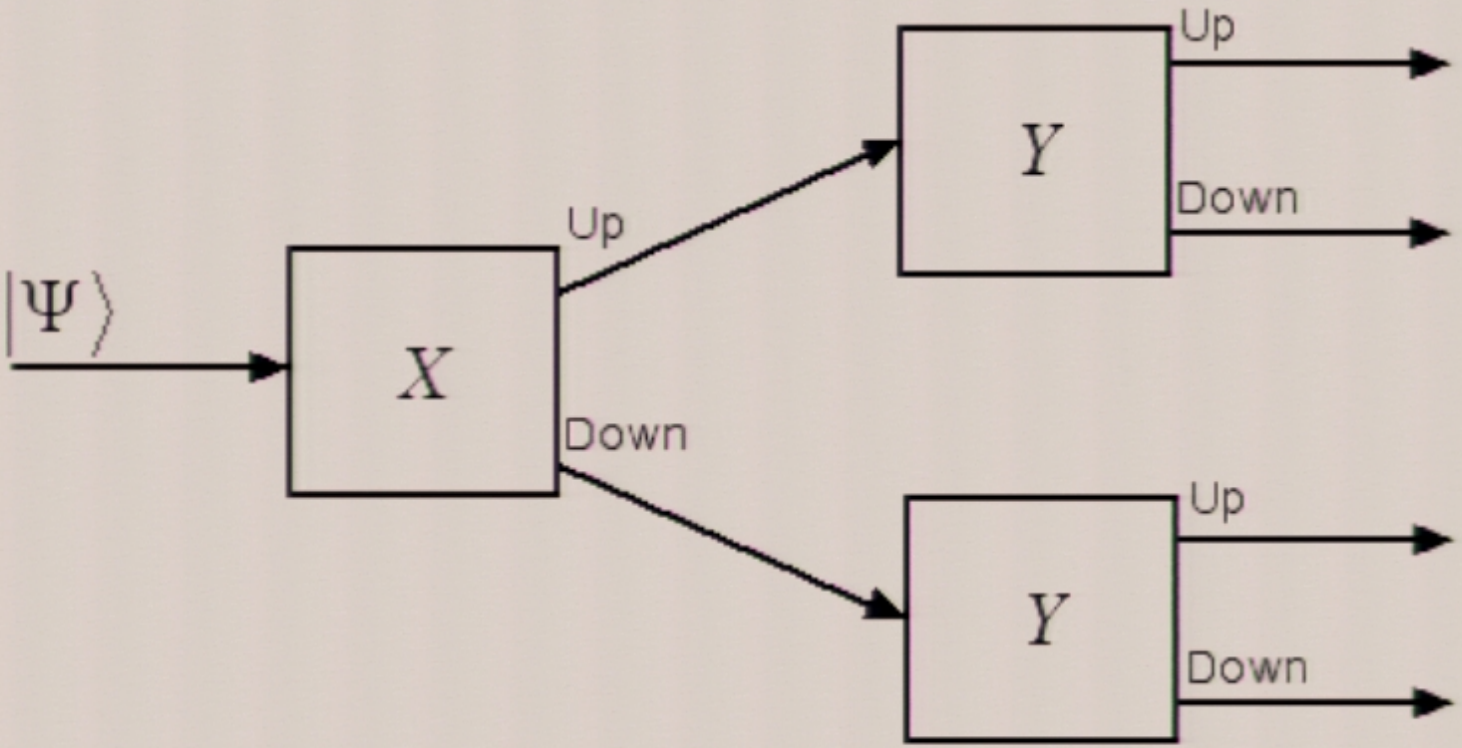
- If $a \in L$ then $a \in S$.
- If $a, b \in S$ then $(a \sqcap b) \in S$.
- If $a \in S$ then $\neg a \in S$.

Not commutative: $a \sqcap b \neq b \sqcap a$

Associative: $(a \sqcap b) \sqcap c = a \sqcap (b \sqcap c) = a \sqcap b \sqcap c$

2) Sequential Quantum Logic

- Developed in late 70's/early 80's by Stachow and Mittlestaedt.
- Extended by Isham et. al. in mid 90's as a possible route to QGravity.



2) SQL: Structure

$$\langle L, S, \sqcap, \neg, (,) \rangle$$

L is a set of elementary propositions and S is the set of sequential propositions given by:

- If $a \in L$ then $a \in S$.
- If $a, b \in S$ then $(a \sqcap b) \in S$.
- If $a \in S$ then $\neg a \in S$.

Not commutative: $a \sqcap b \neq b \sqcap a$

Associative: $(a \sqcap b) \sqcap c = a \sqcap (b \sqcap c) = a \sqcap b \sqcap c$

2) SQL: Hilbert Space Model

Hilbert space: $\mathcal{H}$

Elementary propositions: $L(\mathcal{H})$ - projection operators on $\mathcal{H}$.

Notation: $[a]$ - operator associated to prop. a.

Negation: $[\neg a] = I - [a]$ "NOT a".

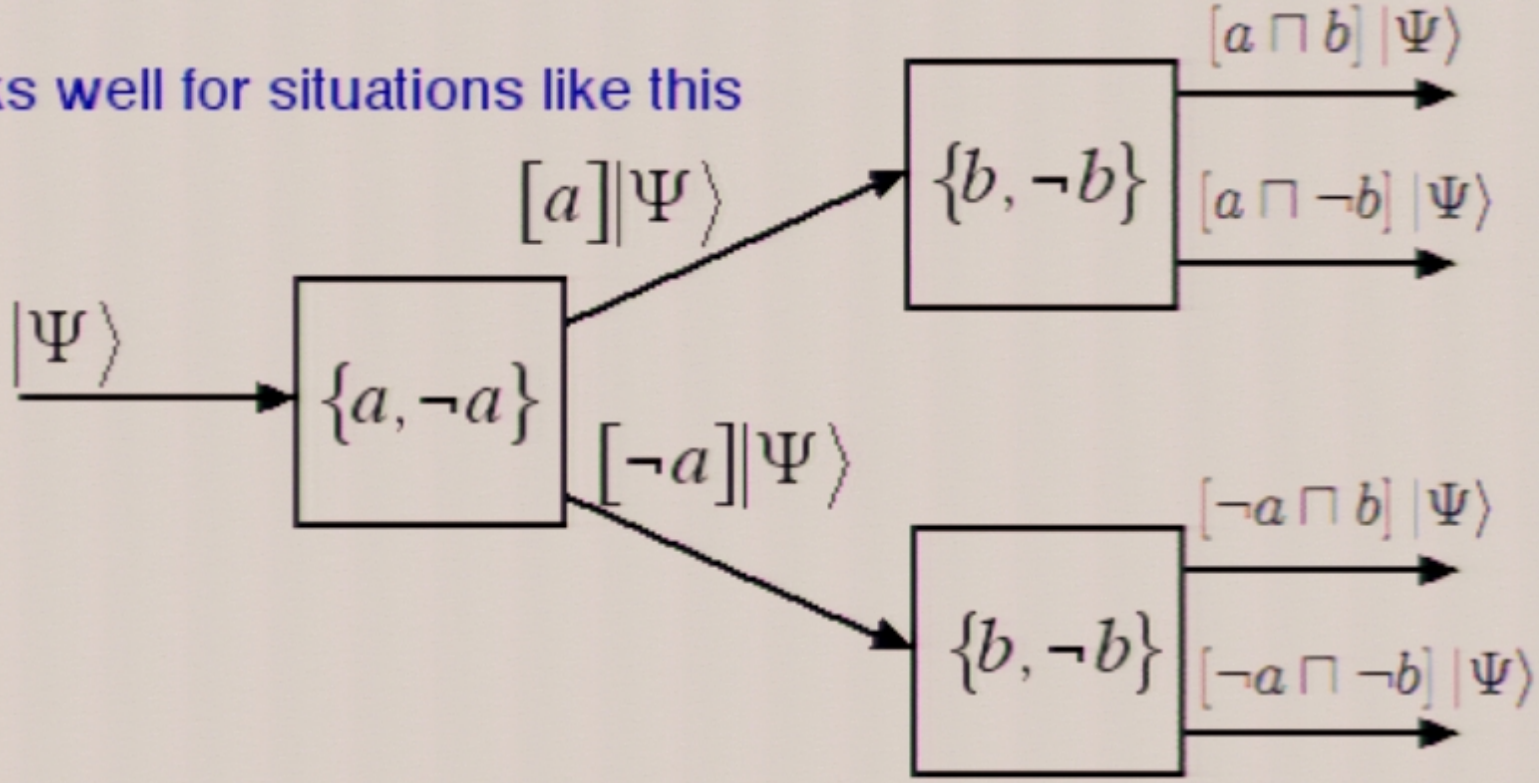
Sequential conjunction: $[a \sqcap b] = [b][a]$ "a AND THEN b".

Note: Usual conjunction can be obtained by including limit propositions

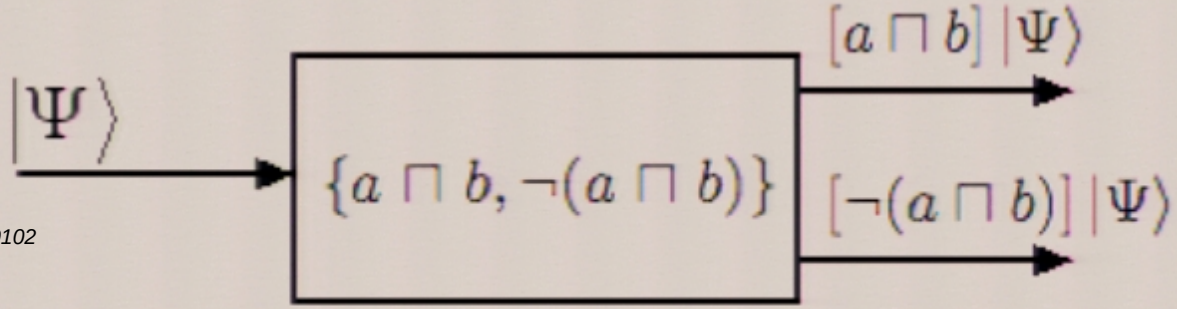
$$[a \wedge b] = \lim_{n \to \infty} ([b][a])^n$$

2) SQL: Problems

SQL works well for situations like this



But it does not handle "coarse-grainings" well:



$$\{[b][a], (I - [b][a])\}$$

2) SQL: Hilbert Space Model

Hilbert space: $\mathcal{H}$

Elementary propositions: $L(\mathcal{H})$ - projection operators on $\mathcal{H}$.

Notation: $[a]$ - operator associated to prop. a.

Negation: $[\neg a] = I - [a]$ "NOT a".

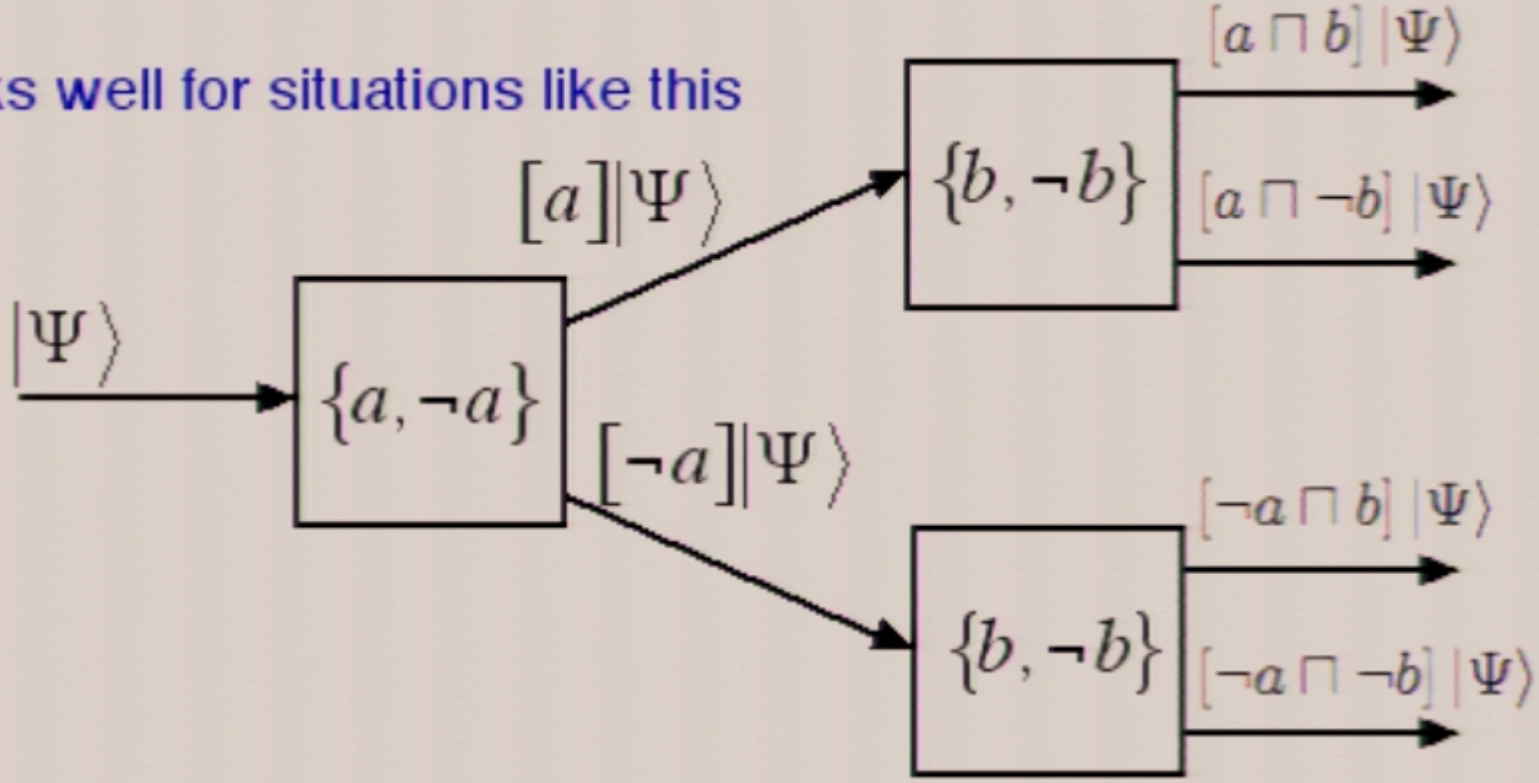
Sequential conjunction: $[a \sqcap b] = [b][a]$ "a AND THEN b".

Note: Usual conjunction can be obtained by including limit propositions

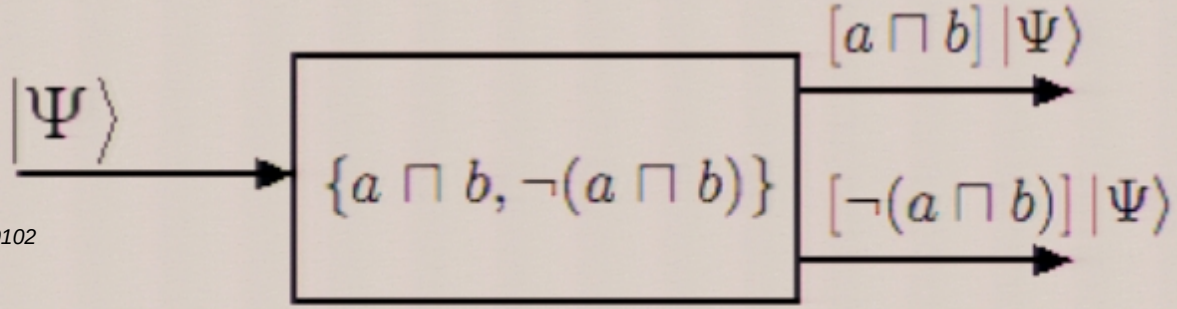
$$[a \wedge b] = \lim_{n \to \infty} ([b][a])^n$$

2) SQL: Problems

SQL works well for situations like this



But it does not handle "coarse-grainings" well:



$$\{[b][a], (I - [b][a])\}$$

3) Testing an SQL Proposition

The algorithm is based on recent QInfo inspired approaches to DMRG.

Cirac, Latorre, Rico Ortega, Verstraete, Vidal, et. al.

It has 3 main steps:

1) Prepare a "history" state that encodes the results of the underlying sequence of measurements.

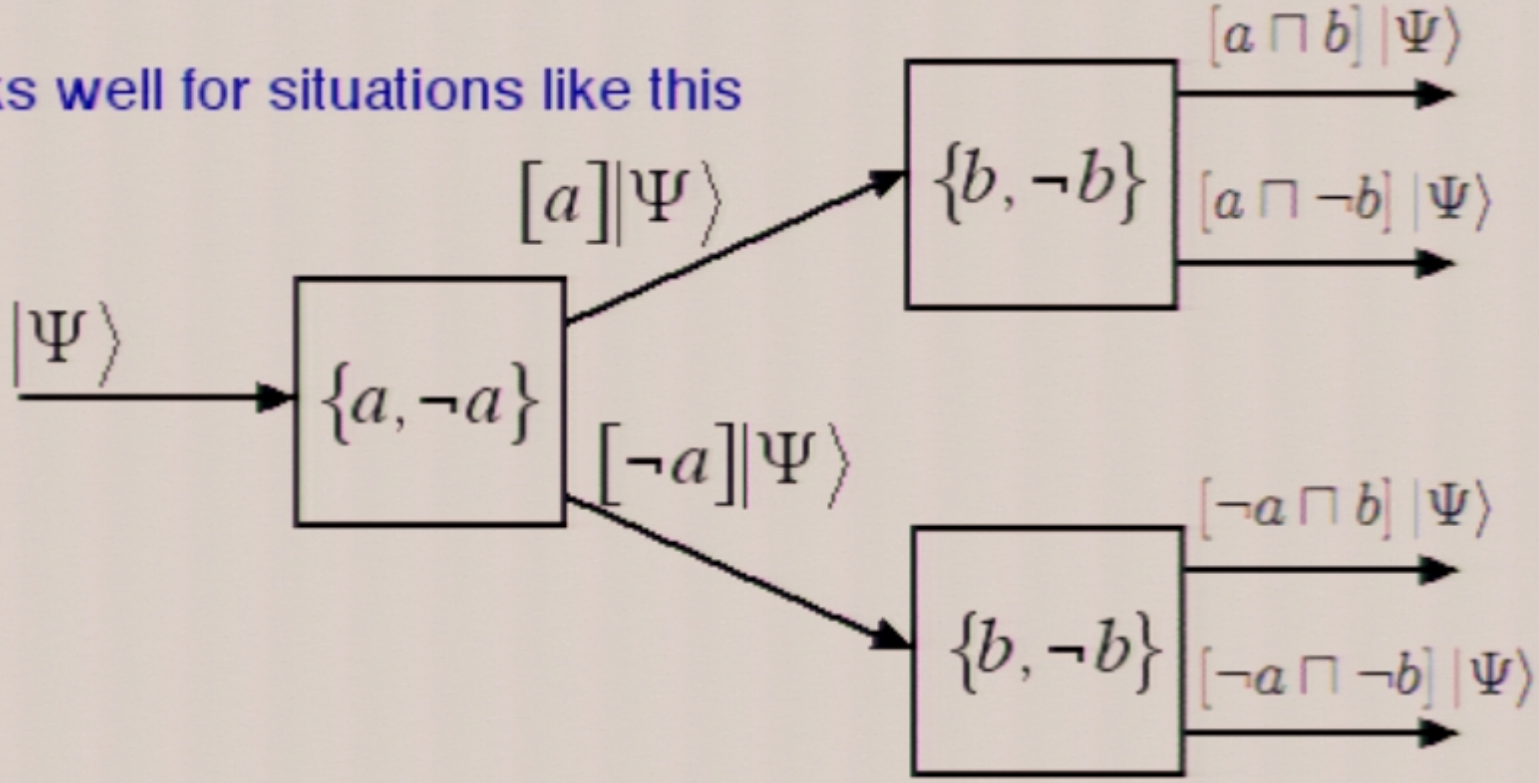
2) Apply rounds of "renormalization" (coherent AND and NOT gates) to get the desired proposition.

3) Measure a qubit to test the proposition.

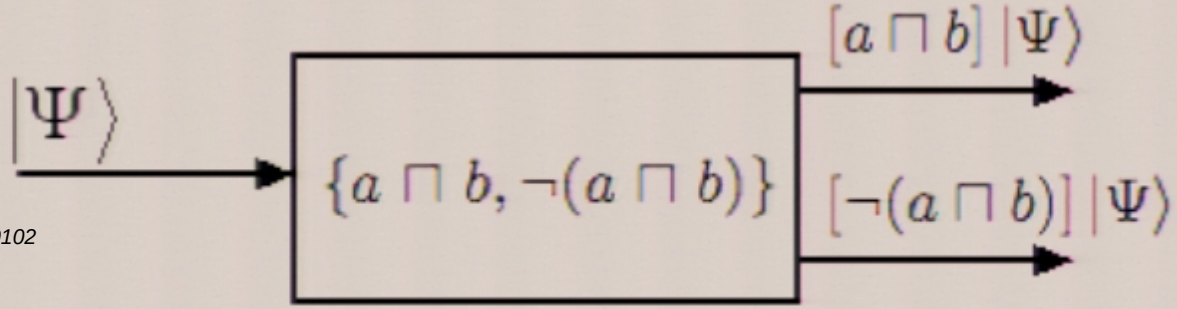
Note: 2) can only be implemented probablistically.

2) SQL: Problems

SQL works well for situations like this



But it does not handle "coarse-grainings" well:



$$\{[b][a], (I - [b][a])\}$$

3) Testing an SQL Proposition

The algorithm is based on recent QInfo inspired approaches to DMRG.

Cirac, Latorre, Rico Ortega, Verstraete, Vidal, et. al.

It has 3 main steps:

1) Prepare a "history" state that encodes the results of the underlying sequence of measurements.

2) Apply rounds of "renormalization" (coherent AND and NOT gates) to get the desired proposition.

3) Measure a qubit to test the proposition.

Note: 2) can only be implemented probablistically.

3) Testing: History state

Suppose we want to test a simple SQL proposition $d = \neg(a \sqcap b) \sqcap c$.

$$x = a, b, c$$

$$[x] = |\psi_x\rangle \langle\psi_x|, \qquad |\psi_x\rangle \in \mathbb{C}^2$$

Notation: $[x^0] = [\neg x], \qquad [x^1] = [x]$

Define: $U_x |0\rangle = |\psi_{\neg x}\rangle, \qquad U_x |1\rangle = |\psi_x\rangle$

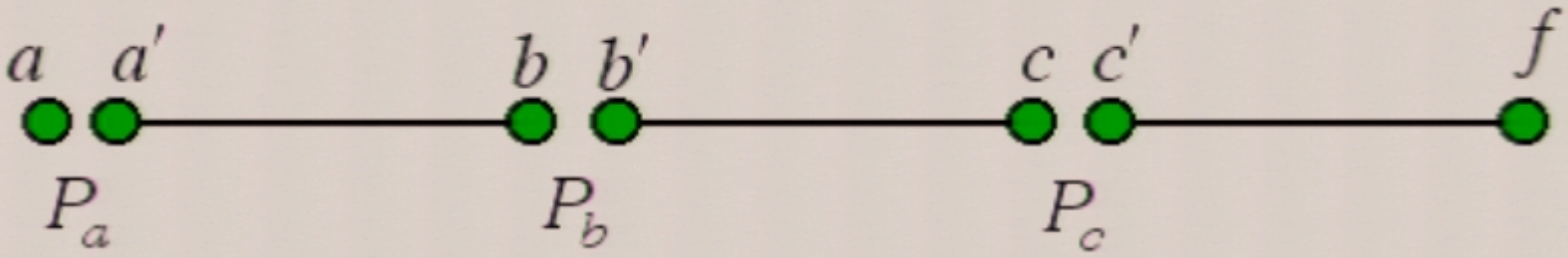
History state: $$\sum_{j,k,m=0}^{1} |j\rangle_a |k\rangle_b |m\rangle_c [c^m][b^k][a^j] |\Psi\rangle_f$$

3) Testing: History state

The history state is a kind of PEPS state.

Starting state:

$$|\text{start}\rangle = |\Psi\rangle_a |\Phi^+\rangle_{a'b} |\Phi^+\rangle_{b'c} |\Phi^+\rangle_{c'f} \text{ where } |\Phi^+\rangle = |00\rangle + |11\rangle$$



$$\sum_{j,k,m=0}^{1} |j\rangle_a |k\rangle_b |m\rangle_c \left[c^m\right]\left[b^k\right]\left[a^j\right] |\Psi\rangle_f = P_a \otimes P_b \otimes P_c |\text{start}\rangle$$

$$P_x = \sum_{j,k,m=0}^{1} \left[x^j\right]_{km} |j\rangle_x \langle km|_{xx'} \qquad [x^0] = [\neg x],\ [x^1] = [x]$$

3) Testing: History state

How to prepare the history state:

i) Apply $U_a^\dagger \otimes U_a^T$ to qubits a and a'.

ii) Perform a parity measurement on a and a'.

$$P_0 = |00\rangle\langle 00| + |11\rangle\langle 11|, \qquad P_1 = |01\rangle\langle 01| + |10\rangle\langle 10|$$

iii) Perform $\mathrm{CNOT}_{a\to a'}$ and discard a'.

iv) If outcome P_1 occurred then perform $U_a X U_a^\dagger$ on qubit b.

v) Repeat steps i) - iv) for (b, b', c) and (c, c', f).

Note: Equivalent to applying operators $\sum_{i,j,k} \left[x^i\right]_{jk} |i\rangle_x \langle jk|_{xx'}$.

3) Testing: Renormalization

$$d = \neg(a \sqcap b) \sqcap c \qquad \sum_{j,k,m=0}^{1} |j\rangle_a |k\rangle_b |m\rangle_c [c^m][b^k][a^j] |\Psi\rangle_f$$

i) Compute $a \sqcap b$ by applying "coherent AND" to qubits a and b.

$$A_{a,b} = |0\rangle_{a\sqcap b} (\langle 00| + \langle 01| + \langle 10|)_{ab} + |1\rangle_{a\sqcap b} \langle 11|_{ab}$$

$$\sum_{j,k=0}^{1} |j\rangle_{a\sqcap b} |k\rangle_c [c^j][(a \sqcap b)^j] |\Psi\rangle_f$$

ii) Compute $\neg(a \sqcap b)$ by applying X to qubit $a \sqcap b$.

$$\sum_{j,k=0}^{1} |j\rangle_{\neg(a\sqcap b)} |k\rangle_c [c^j][\neg(a \sqcap b)^j] |\Psi\rangle_f$$

iii) Compute d by applying $A_{\neg(a\sqcap b),c}$.

$$\sum_{j=0}^{1} |j\rangle_d [d^j] |\Psi\rangle_f$$

3) Testing: Implementing AND

$$A_{a,b} = |0\rangle_{a\sqcap b} \left(\langle 00| + \langle 01| + \langle 10|\right)_{ab} + |1\rangle_{a\sqcap b} \langle 11|_{ab}$$

is not a directly implementable. Instead, implement measurement:

$$M_s = \tfrac{1}{\sqrt{3}} \left(|01\rangle \left[\langle 00| + \langle 01| + \langle 10|\right] + |11\rangle \langle 11|\right)$$

$$M_f = \left(I - M_s^\dagger M_s\right)^{1/2}$$

If M_s, discard 2nd qubit and proceed.

If M_f, abort and restart algorithm from beginning.

Note: Algorithm will succeed with exponentially small probability in no. $\sqcap$ gates.

4) Generalizations

- Testing projectors of arbitrary rank.
- Testing props on d-dimensional Hilbert space.
  - Need upper bound on d needed to get correct probs. for all formulae of length n.
- Testing multiple propositions.
  - On disjoint subsets of an underlying sequence of propositions.
  - By copying qubits in the computational basis.

6) Open Questions

Can SQL be modified so that all sequential propositions can be tested?

Modify definition of sequential conjunction.

Restrict allowed subspaces.

Is SQL the logic of an interesting model of computing?

c.f. Aaronson's QC with post-selection.

Renormalization: A new paradigm for irreversible quantum computing?

Which DMRG schemes are universal for classical/quantum computing?

Is there a natural quantum logic which has SAT problems that are NQP or QMA complete?