e.g. $W = \mathbb{F}$:

$$\Lambda(X) = tr_B\left(\mathbb{F}(I_A \otimes X_B)\right)^T = tr\left(I_A \cdot X_B\right)^T = X_B^T.$$

$\Rightarrow$ PPT criterion!

Note: PPT strictly stronger: $\mathbb{F}$ could not detect e.g. $|\Psi\rangle$!

Corollary: A state is entangled if & only if

$(\Lambda \otimes I)(\rho) \geq 0$ $\forall$ positive Λ (as sep. states $\Leftrightarrow$ all witness)

e) Quantification of mixed state entanglement

How to quantify entanglement?

i) Entanglement needed to create state

"Entanglement of formation" E_F (single copy)

"Entanglement cost" E_C (many copies)

ii) Extractable entanglement:

"Distillable entanglement" E_D:

LOCC protocol $\mathcal{E}_n$: $\| \mathcal{E}_n(\rho^{\otimes n}) - |\Omega\rangle\langle\Omega|^{\otimes n} \| \to 0.$

Note: Generally $E_C \neq E_D$: no unique measure!

Problem: E_F very hard, E_c / E_D (almost) impossible to compute $\Rightarrow$ need other measures.

(But: Cases w/ $E_D = 0$ known, e.g. PPT states. Currently big open problem.)

Have seen: ρ^{T_A} has neg. eigenvalues $\Rightarrow \rho$ entangled.

Use as ent. measure:

$$\underline{\text{Negativity}} \quad N(\rho) = \frac{1}{2}\Big(\underbrace{\sum_i |\lambda_i(\rho^{T_A})|}_{=: \|\rho^{T_A}\|_1} - 1\Big)$$

$$= \frac{1}{2}\left(\|\rho^{T_A}\| - 1\right) = \sum_{\substack{\text{neg.} \\ \text{eigenvals}}} \left(-\lambda_i(\rho^{T_A})\right).$$

or $\underline{\text{log-negativity}} \quad E_N(\rho) = \log_2 \|\rho^{T_A}\|_1$

What are desired properties for ent. measure E?

- $E_D \le E \le E_c$.
- 0 on sep. states, $\neq 0$ on ent. states.
- additive: $E(\rho_{AB} \otimes \sigma_{A'B'}) = E(\rho_{AB}) + E(\sigma_{A'B'})$
- LOCC-monotone: cannot be increased by LOCC.
- Coincides with $E(|\psi\rangle) = S(\mathrm{tr}_B |\psi\rangle\langle\psi|)$ for pure states.

Negativity / Log-negativity:

$\mathcal{N}$: • LOCC-monotone

- 0 on sep. states, but can be $\neq 0$ on ent. states.
- $= E(|\psi\rangle)$ for pure
- not additive

E_N : • additive

- 0 on sep., but can be $\neq 0$ on ent. states
- $= E(|\psi\rangle)$ for pure
- not an LOCC monotone.

IV. Quantum Computation

1. The circuit model

a) Classical computation

Task of class. computers:

Solve problems ≡ compute functions:

$$f: \{0,1\}^n \longrightarrow \{0,1\}^q$$

$$\underline{x} = (x_1, \dots, x_n) \longrightarrow f(x_1, \dots, x_n)$$

f depends on problem, $\underline{x}$ encodes instance of problem.

E.g.: Multiplication: $(a,b) \mapsto a \cdot b$

$$\underline{x} = (\underbrace{\underline{x}_1, \underline{x}_2}_{\text{binary enc. input}}) \Rightarrow f(\underline{x}) = \underbrace{\underline{x}_1 \cdot \underline{x}_2}_{\text{in binary}}$$

Factorization:

$\underline{x}$: integer; $f(\underline{x})$: list of prime factors (w/ suitable encoding)

Each problem is encoded by a family of functions

$f = f^{(n)}: \{0,1\}^n \to \{0,1\}^m$; $m = \text{poly}(n)$, $n \in \mathbb{N}$

Must be possible to "construct f systematically & efficiently"
(→ later!)

What ingredients do we need to compute a general function f?

(i) $f: \{0,1\}^n \to \{0,1\}^m$

$$f(\underline{x}) = (f_1(\underline{x}), f_2(\underline{x}), \ldots, f_m(\underline{x}))$$

$f_k: \{0,1\}^n \to \{0,1\}$

⇒ can restrict to Boolean functions $f: \{0,1\}^n \to \{0,1\}$.

(ii) Let $L = \{\underline{y} \mid f(\underline{y}) = 1\} = \{\underline{y}^1, \underline{y}^2, \ldots, \underline{y}^k\}$

Define $g_{\underline{y}}(\underline{x}) = \begin{cases} 0 ; & \underline{x} \neq \underline{y} \\ 1 ; & \underline{x} = \underline{y} \end{cases}$

$$f(\underline{x}) = g_{\underline{y}^1}(\underline{x}) \vee g_{\underline{y}^2}(\underline{x}) \vee \ldots \vee g_{\underline{y}^k}(\underline{x})$$

"$\vee$": logical "or": $0 \vee 0 = 0$, $0 \vee 1 = 1$, $1 \vee 0 = 1$, $1 \vee 1 = 1$

associative

⇒ $a \vee b \vee c = a \vee (b \vee c)$ etc!

(iii) $g_y(\underline{x}) = (y_1 = x_1) \wedge (y_2 = x_2) \wedge \dots$

"$\wedge$": logical "and": $1 \wedge 1 = 1$

otherwise $x \wedge y = 0$

(iv) $(y_i = x_i) = \begin{cases} x_i, & y_i = 1 \\ \neg x_i, & y_i = 0 \end{cases}$

"$\neg$": logical "not": $\neg 1 = 0$

$\neg 0 = 1$

$\Rightarrow$ $f(\underline{x})$ can be built from 4 predicates:

"and", "or", "not" gates, + copy gate $x \mapsto (x, x)$.

"Universal gate set".

(Note: In fact, either $\neg(x \wedge y)$ "nand" or $\neg(x \vee y)$ "nor", together w/ copy, are already universal!)

Circuit model of computation:

1 built from universal gate set (without "loops such in time").

Which problems can we solve efficiently w/ this model?

⇒ Problems where # gates (= # "time steps") scales nicely with n, i.e. as some polynomial $\text{poly}(n)$. (Class "P" of problems.)

Otherwise: hard problem.

Is a typical problem easy or hard?

$$f: \{0,1\}^n \to \{0,1\}$$

different f: $2^{(2^n)}$ ← # inputs

(on each input: 0 or 1)

But only $c^{\text{poly}(n)}$ circuits of length $\text{poly}(n)$

↑ # elementary gates

⇒ most f cannot be computed efficiently.

Note: Also require f to be a uniform family of circuits: can e.g. be generated by a computer program from input n in time $\text{poly}(n)$ for any n.

Does comp. power dep. on gate set?

→ No. By def., any universal gate set can simulate any other w/ constant overhead; same power!

What about other models of computation?

- CFG
- parallel computer
- Turing machine (tape based)
- Cellular automata

⋮ (other exotic models...)

⇒ All known "reasonable" models can simulate each other w/ poly(n) overhead ⇒ same comput. power!

Church-Turing-Thesis: All reasonable models of computation have the same computational power.

(Note: different variants; randomness? quantum ("strong" C-T-Thesis))

Will use circuit model to go to quantum systems:

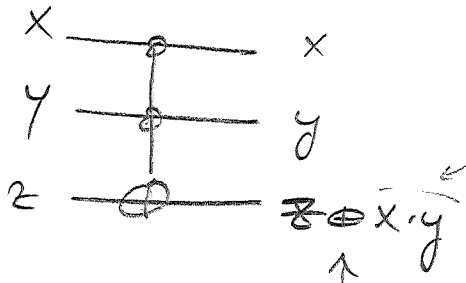
Gates → Unitaries

Rmk: Class. gates irreversible ⇔ unitaries reversible.

Can we fit even class. comput. in this picture?

Classical computation can be turned reversible:

Toffoli gate:



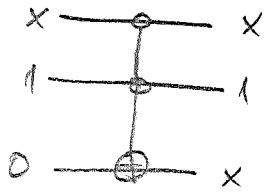
$x, y, z \in \{0,1\}$

sum mod 2 = "XOR"

$0 \oplus 0 = 0$
$0 \oplus 1 = 1$
$1 \oplus 0 = 1$
$1 \oplus 1 = 0$

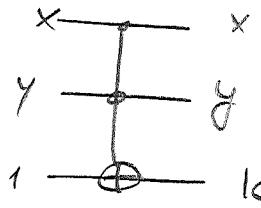
→ reversible!

→ can simulate and/or/not/copy using ancillas, e.g.:



COPY

;



⇒ reversible univ. gate (using ancillas)

Any $f(x)$ can be computed reversibly:

$$f^R : (x, y) \longmapsto (x, f(x) \oplus y)$$

($\oplus$ bitwise XOR.)

- possibly w/ ancillas - w/out changing anything else.

(Idea: Compute f reversibly w/ ancillas, XOR result w/ y, and "un-compute" everything - clean ancillas.
— Can be optimized to use few ancillas! → Prakti)

⇒ Everything can be computed reversibly.

But: 3-bit gate needed (→ HW)

5) Quantum Circuits

Model of quantum computation:

- System consists of qubits (or d-its): tensor prod. structure
- Choose universal gate set

$$S = \{U_1, \ldots, U_k\} \text{ of "small" (i.e. few-qubit) gates}$$

⇒ Sized (poly-size) circuits.