**Domande QUANT**

**Fisica della computazione e modelli alternativi**

**1- Parla delle porte logiche, di quelle universali e della reversibilità**

Pur formando dei set universali, le porte AND, OR, NAND e NOR sono però irreversibili. Questo perchè ricevono due bit di input ma generano un solo bit di output. Dunque parte dell’informazione iniziale viene persa. Esiste una porta logica che è allo stesso tempo reversibile e universale. Venne proposta nel 1982 da Fredkin e Toffoli. Essa può essere interpretata come uno switch controllato di bit. Il bit di controllo è A; se questo è acceso i bit B e C vengono scambiati altrimenti vengono lasciati identici.

**2- Parla delle operazioni bit-a-bit**

Consideriamo uno spazio logico generato da n bit. A una stringa di n bit possiamo associare un intero compreso fra 0 e N - 1 (associati rispettivamente alle stringhe 000...00 e 111...11) con N = 2n. In genere, è possibile passare da un intero alla una stringa di n bit associando la stringa di bit x1x2...xn con xi = 0, 1 e i = 0, 1, ..., n tale che x = x1\*2n-1 + ... + xn\*20

**3- Parla brevemente del computer a biliardo**

v. slide hahaha

**4- Parla brevemente del computer a dna e del problema Hamiltoniano**

v. slide hahaha

**Apparato matematico**

**5- Definisci prodotto tensore e sue proprietà. Notazione braket?**

Consideriamo ora due spazi vettoriali V e W con le loro basi. Deduciamo che dim(V) = n e dim(W) = m. Il prodotto tensore dei due spazi viene indicato con V ⌦ W e ha dimensione dim(V ⌦ W) = n \* m. Le sue proprietà sono: distributiva (sx e dx) e distributiva rispetto ad uno scalare. Se V e W ammettono prodotto scalare, anche V ⌦ W lo ammetterà.

Nella notazione braket un vettore (che poi sarà il nostro stato quant.) è rappresentato come bra: <a| (vettore trasposto (riga)): o come ket: |a> (se ortonormali ho: <v,v>=1 e <v,w>=0)

Abbiamo che <w, u> = 0 se w ortog. a u, e che <u, u> = |u|2 (norma al quadrato)

Il prodotto scalare ha le seguenti proprietà: <u, u> = 0 sse u = 0 ; <v, w> = (<w, v>)\* con \* = complesso coniugato ; <w, au + bv> = a<w, u> + b<w, v>.

**6- Parla degli operatori, con notazione braket e matriciale. Hermitiani?**

Un operatore è una trasformazione del tipo o:v→v

Data una base v scriviamo: Oij = <ai|O|aj> e costruiamo una matrice con le entrate i,j

Ad es. se O = |w><v| ho O|v> = (|w><v|)|v> = (|w>)<v||v> = |w>\*1 = |w>

Si può scrivere O come sommatoria per k,m di Okm \* |ak><am| e osservare che, data la linearità, sviluppando <ai|O|aj> rimane solo il termine Okm \* dirac(ik)\*dirac(mj) = Oij

Se O = O+ si dice che O è hermittiano.

**7- Parla di autovalori e autovettori. Fai esempi di esercizi.**

Se O|v> = lamda\*|v> ho che lambda è EW di O e |v> è EV di O (equazione agli autovalori).

Inoltre, se O è hermittiano, si può dimostrare che gli autovalori sono reali.

**Fenomeni quantistici**

**8- Parla dell’esperimento della doppia fenditura**

v. slide hahaha

**9- Parla dell’esperimento della luce polarizzata**

v. slide hahaha

**10- Cos’è un sistema quantistico? Cos’è il principio di sovrapposizione?**

Un sistema fisico a due stati è descritto da uno stato quantistico |ai = a0 |v0> + a1 |v1>, ovvero da una combinazione lineare di due stati, |v0> e |v1>, che costituiscono una base ortonormale di uno spazio vettoriale complesso V. Le ampiezze a0 e a1 soddisfano il vincolo |a0|2 + |a1|2 = 1, per cui il vettore che rappresenta uno stato ha norma unitaria. Lo spazio V è detto spazio degli stati. Il fatto che uno stato quantistico sia una combinazione lineare a coefficienti complessi di stati è un postulato noto come principio di sovrapposizione degli stati. Un sistema quantistico con due stati può essere pensato e utilizzato come un quantum bit o qubit. La dizione sistemi fisici a due stati non deve trarre in inganno: gli stati possibili in cui si può trovare un sistema fisico a due stati, infatti, sono infiniti!

**11- Cos’è la misura? E il collassamento dello stato?**

Se il nostro sistema quantistico è descritto dallo stato |ai> = a0 |v0> + a1 |v1> nella base {|v0>, |v1>} e con |a0|2 + |a1|2 = 1, il postulato della misura ci dice che una misura detta proiettiva darà come risultato il valore associato allo stato |v0> con probabilità |a0|2 e il valore associato allo stato |v1> con probabilità |a1|2. Inoltre, la misura proiettiva distrugge la sovrapposizione degli stati. Il secondo punto del postulato della misura dice che dopo la misura il sistema collassa nello stato associato al valore ottenuto. Se dopo la prima misura il sistema è collassato in un certo stato, una qualsiasi misura successiva nella stessa base darà sempre lo stesso risultato. Un osservabile fisico (ad esempio, la polarizzazione) può essere associato ad un operatore Hermittiano. Gli operatori Hermittiani possono essere diagonalizzati e possiamo ottenere i loro autovalori e autovettori. Se vogliamo misurare l’osservabile A, il postulato della misura ci dice che i risultati della misura possono essere solo gli autovalori e il sistema collasserà nell’autostato rispettivo dell’operatore A.

La base, gli autovalori e il risultato della misura dipendono dall’osservabile.

**12- Riassumi in 3 punti le regole del postulato della misura.**

1. Se vogliamo misurare un osservabile A, dobbiamo conoscerne i suoi EW e EV; cioè gli stati tali che A|EVi> = EWi\*|EVi>. Gli autovettori saranno la base su cui decomporre lo stato del nostro sistema. Ovvero dobbiamo scrivere |a> = SUM(ai \* |EVi>) con ai = <EVi | a>

2. La misura avrà come risultato l’autovalore i con probabilità |ai|2.

3. Successivamente alla misura, il sistema si troverà nello stato associato all’EW misurato.

**13- Dai un’interpretazione quantistica dell’esperimento con la luce polarizzata**

Se il sistema è in uno stato |v0> per sapere quali possono essere i risultati della misura di polarizzazione dobbiamo scrivere lo stato come combinazione lineare degli autostati di P {|up>, |right>}. Una misura di P ci darà il valore +1 con prob ad es. 75% e il sistema collasserà in |up> (nel rimanente 25% dei casi sarà -1 e collasserà nello stato |right>).

Quando diciamo che un fotone è in uno stato quantistico dato dalla sovrapposizione di due stati non intendiamo che il fotone è nello stato 1 con probabilità x e nello stato 2 con probabilità y, ma che è in uno stato che è in una delle infinite possibili combinazioni lineari di 1 e 2. Non ha senso chiedersi in che stato sia il fotone se non effettuando una misura, il risultato della quale sarà sempre e solo uno di due possibili.

**14- Parla della fase globale e di quella relativa. Cosa varia?**

Dati i vettori |u> e eix |u> che hanno lo stesso modulo ma differiscono per una fase globale, il calcolo delle probabilità dei risultati di una qualunque misura fornisce sempre gli stessi valori. La fase globale non è dunque misurabile perchè non esiste un osservabile che possa distinguerla in fase di misura. Al contrario, se la fase relativa (ad esempio |+> e |->) è possibile trovare un osservabile che li distingua.

**15- Come formalizzo gli stati a n q-bit?**

Formalizziamo lo stato a n-quibt (con dim 2^n) tramite il prodotto tensore v0 x v1 x …x v(n-1). Come si può vedere gli stati che compongono la base del sistema a n qubit rappresentano una base logica delle stringhe a n bit. I ket della base canonica saranno scritti come vettori a 2^n componenti in cui tutti gli elementi sono nulli tranne uno.

**16- Quando uno stato a due q-bit è separabile?**

Gli stati che si possono scrivere come il prodotto tensore |v> ⌦ |w> vengono detti fattorizzabili. Si può dire che gli stati del sistema V e W sono "separabili" e non si influenzano a vicenda durante un processo di misura. Se scriviamo |v> ⌦ |w> = (a|0> + b|1>) ⌦ (c|0> + d|1>) vediamo dunque che (riscrivendolo) a|0>(c|0> + d|1>) + b|1>(c|0> + d|1>), dopo una misura del primo qubit collassa in |0> o |1> (di V), mentre W a seguito di tale misura si trova indipendentemente nello stato di sovrapposizione (c|0> + d|1>)

**17- Quando invece non lo è? Cosa sono gli stati entangled? A cosa servono?**

Se invece scrivo (|0>v |1>w + |1>v |0>w)/sqrt(2) = (|00>+|11>)/sqrt(2) NON posso fattorizzarlo come prodotto tensore di |v> e |w> (vive nel prodotto tensore, ma non è separabile dato che dovrei avere ac = bd = 0 ma anche ad = bc = 1/sqrt(2) (no sol.)

**18- Quali sono gli stati di Bell?**

|00> + |11>, |00> - |11>, |01> + |10>, |01> - |10> (normalizzati e orotonorm. tra loro)

**19- Quali buone proprietà hanno le trasformazioni unitarie?**

Le trasformazioni unitarie sono lineari e invertibili (U-1 = U+) e preservano il prodotto scalare:

Dato |v1> = U|u1> e <v1| = <u1|U+ ho <v1|v2> = (<u1|U+)(U|u2>) = <u1|u2>

**20- Quali sono le trasformazioni di Pauli?**

In base B1:

X= |0X1| + |1X0|, Y= i \* |0X1| - i \* |1X0|, Z = - |0X0| + |1X1|

In base B2:

Z= |+X-| + |-X+|, Y= i \* |+X-| - i \* |-X+|, X = |+X+| - |-X-|

**21- Come posso scrivere una trasformazione generica?**

U = (cos(a/2) \* Id + i\*sin(a/2))(nX + mY + kZ)

**22- Cos’è la trasformazione di Hadamard?**

Serve a passare da una rappresentazione con Z diagonale a una in cui X è diagonale (cioè a cambiare base). Ad es. H|0> = |+>.

Posso scriverla come 1/sqrt(2)\*((1, 1), (1, -1))

**23- Cos’è il Controlled-NOT? Perché è importante?**

I CNOT servono a creare e distruggere entanglement, negando il secondo qubit se il primo è 1 (ad esempio |10> → |11>). Ad es. |00> con l’applicazione di H1 e CNOT(1,2) mi da il primo stato di Bell (ho creato entanglement).

**24- Come calcolo i valori medi degli operatori di Pauli?**

Dato |phi> = cos(a/2)|0> + sin(a/2)\*ei\*phi|1> e <phi| = cos(a/2)<0| + sin(a/2)\*ei\*-phi<1|

Abbiamo: X : <phi|X|phi> = … = sin(a)cos(phi), Y : … = sin(a)sin(phi), Z: … = cos(a)

**25- Fai il parallelismo con la sfera di Bloch**

I due angoli a e phi possono essere usati per parametrizzare un punto su una sfera di raggio unitario. Phi è l’angolo sul piano X&Y, e a sul piano Z.

**Informazione quantistica**

**26- A cosa corrispondono le varie rotazioni sulla sfera di Bolch?**

Con U generica ho una trasformazione generica, con X ho una rotazione sull’asse X ecc

**27- Cosa deduco dal teorema di Eulero?**

Dal teorema di Eulero deduco che mi bastano due dei tre operatori di Pauli per esprimere tutte le rotazioni possibili.

**28- Perché posso racchiudere tutta l’informazione in un signolo q-bit?**

Applicando una porta di Hadamard costruisco un qubit in sovrapposizione di stati logici.

Questa osservazione è alla base del parallelismo quantistico, grazie al quale posso codificare tutta l’informazione logica, cioè tutte le possibili stringhe a n bit.

**29- Che accade se applico due porte di Hadamard ai singoli q-bit?**

Lo stato ottenuto applicando due porte di Hadamard è sovrapposizione di tutti i possibili stati logici. Se adesso applichiamo un operatore unitario U, ricordando le proprietà di linearità della meccanica quantistica, otterremo che U agisce su tutti i singoli qubit parallelamente.

**30- Fai considerazioni su pro e contro della parallelizzazione.**

Se da un punto certo punto di vista la meccanica quantistica permette una manipolazione parallela, l’estrazione dell’informazione è estremamente complicata (probabilistica)

**31- Parla del teorema del no-cloning e dimostralo**

Non può esistere un operatore generico che copi uno stato quantistico in un altro. Ad esempio il CNOT funziona ma solo per la base 0,1.

Dimostrandolo (v. slide) si nota che posso copiare solo stati uguali o ortogonali tra loro.

**32- Parla del super-dense coding: premesse e risultati**

Supponiamo A e B debbano scambiarsi dei bit di informazione. In particolare, A vuole mandare due bit di informazione classica. Se Alice e Bob possono usare la meccanica quantistica la stessa procedura può essere completata spedendo un singolo qubit. A e B devono condividere uno stato entangled e la procedura è composta da:

A: applicazione di una porta → invio a Bob → B: CNOT → H

**33- Parla del teletrasporto quantistico: come funziona? Osservazioni?**

A e B devono condividere uno stato entangled e la procedura è composta da:

A: partendo da |phi>\*|Bell> → CNOT → H → misura con collasso in |00>,...|11> con probabilità ¼, che se interpretati come bit classici possono essere inviati a Bob cosicchè lui possa applicare la corretta porta logica associata e ottenere il giusto stato.

Oss1: non posso inviare pigreco (! discretizzabile): cioè posso, ma non posso misurarlo

Oss2: Bob non può estrarre informazioni sui coefficienti L’informazione cioè non può essere usata da Bob fino a che Alice non gli comunica il risultato della sua misura.

**34- Teletrasporto quantistico: dimostra che Bob non può estrarre info**

Se Bob facesse una misura sul suo qubit otterrebbe la metà delle volte 0 e l’altra metà 1. Supponiamo che A non comunichi a Bob il suo risultato. B non può avere informazioni (statistiche) sullo stato che Alice gli voleva mandare perchè:

* se alice ha misurato 00 Bob avrà p = |a|^2 di misurare 0 e |b|^2 di misurare 1
* se alice ha misurato 01 Bob avrà p = |b|^2 di misurare 0 e |a|^2 di misurare 1
* … altre due volte

→ ho ¼(2\* |a|^2 + 2\* |b|^2) = ½ che era la p dello stato entang. iniziale senza considerare le azioni di Alice

**35-39 Deutsch, Deutsch-Jozsa, Vazirani, Simon: 📦**

**Crittografia Quantistica**

**38- Qual è lo scopo del Quantum Key Distribution? Quali premesse in BB84?**

Lo scopo è distribuire la chiave privata. BB84 è un protocollo sicuro per la distribuzione di chiavi crittografiche. L’idea è che A e B possano scambiarsi una chiave crittografica sicura con cui criptare il messaggio. La meccanica quantistica ha due proprietà fondamentali che permettono la sua implementazione: Un eventuale hacker non può copiare un generico qubit contenente l’informazione a causa del teorema no-cloning in sec. 2. una misura del qubit lo perturba a causa del collasso della funzione d’onda. L’idea alla base del protocollo BB84 è che dato uno stato quantistico esiste una base in cui l’output della misura è certo e non probabilistico.

**39- Esponi i primi 6 passi di BB84**

1) A: estrazione n qbit random (bit logici)

2) A: estrazione n qbit random (basi)

3) Dati |->=|1+> e |+>=|0+> A associa B1 a 0 e B2 a 1 → A invia a B

4) B: estrazione n qbit random (basi) e misura i qubit ricevuti nelle basi estratte

5) A, B pubblicano le basi: agli altri non da informazioni, ma A e B possono confrontare le basi tra di loro e, dove coincidono, sapere che il bit in quella posizione è al 100% quello.

6) Avrò così n/2 bit correlati → di solito quindi invio 2\*key\_len che voglio (o + se rumore)

**40- E se comparisse un Attacker E?**

Il massimo che può fare Eve è estrarre (come farebbe B) le basi random e sostituirsi a Bob tra il passo 3 e 4. Può dunque misurare i bit nelle basi estratte e ricadere nei seguenti casi:

* le basi di A e B sono diverse tra loro → ignoreranno a prescindere
* le basi di A e B coincidono:
  + Eve indovina la base giusta (Eve passa inosservata)
  + Eve non indovina la base giusta

Molte volte Eve non riesce a sostituirsi a Bob (Eve misura nella base sbagliata ottenendo una perturbazione → Bob se ne accorgerà)

**41- Come risolvere l’attacco di Eve? Inoltre, n è pubblico?**

Affinchè A e B possano tra di loro verificare intrusioni pubblicano metà della chiave (metà degli n/2 che risultavano correlati = n/4 → uso quindi n\*4 bit se voglio key\_len = n): la presenza di un qubit diverso laddove le basi dovrebbero essere uguali indica l’intrusione di Eve (che in alcuni casi ha perturbato misurando nelle basi sbagliate). La perturbazione di Eve è statisticamente maggiore di quella eventualmente operata dal rumore.

**42- Come gestire eventuale rumore?**

Se c’è del rumore si stabilisce una soglia di correlazione tra A e B per distinguerlo da Eve.

**43- Esponi EKERT91**

Per questo protocollo si utilizza l’entanglement (dato lo stato di Bell |phi+>):

A e B estraggono 0 o 1, indicanti la base in cui misurare: se le basi coincidono, dato l’entanglement, dopo la misura di uno dei due (a cui segue il collasso in |00> o |11> (o |++> o |-->) con probabilità ½) la misura successiva sarà al 100% uguale a quella precedente.

Se invece la base scelta da A è diversa da quella di B, la seconda misura perturberà generando una scorrelazione (ad es. se dopo aver misurato |00> (associato a EW 0) misuro nella base B2 posso collassare in |0+> o |0-> e ho dunque ½ di prob di trovare EW 1).

Eve può provare ad inserirsi (con le stesse considerazioni di BB84) ma solo durante lo scambio dei qbit entangled (perchè manca una fase lungo un canale come BB84)

**Grover**

**44- Su che considerazioni si basa l'algoritmo di Grover?**

Il primo passo per arrivare all’algoritmo per la ricerca in database è introdurre il concetto di oracolo. Supponiamo di avere un database di N elementi. Invece di classificali e distinguerli in base all’informazione che contengono, a livello astratto, è utile associare ad ogni elemento un numero intero. La ricerca nel database si ridurrà quindi a trovare l’intero che soddisfa le proprietà desiderate. Per definizione, f(x) = 1 se x è soluzione del nostro problema (ovvero è uno degli elementi che stiamo cercando) e f(x) = 0 se x non è soluzione del nostro problema. L’oracolo non conosce la soluzione del problema ma sa riconoscerla quando viene interrogato sottoponendogli un elemento del database. La seconda proprietà che deve avere l’oracolo è di poter marcare la soluzione del problema (cambiando il segno)

**45- Esponi i passi dell'algoritmo**

1) L’algoritmo di Grover inizia con la costruzione dello stato quantistico sovrapposizione di tutti i possibili stati logici (applico n porte di Hadamard a |0..0>)

2) Applicazione operatore di Grover:

oracolo O che cambia la fase agli stati soluzione → n porte di Hadamard → cambio

fase a tutti gli stati tranne a |0..0> → n porte di Hadamard

3) ripeto O(sqrt(N)) volte

**46- Come dividiamo gli stati soluzione e non?**

Lo spazio logico può essere diviso in due sottospazi. Quello generato da |x> con x soluzione del nostro problema (ovvero quelli a cui l’oracolo cambia segno) e quelli che non sono soluzione del nostro problema. Supponiamo che ci siano M stati soluzione e, di conseguenza N - M stati non-soluzione:

avremo |a> = 1/sqrt(N-M)\*SUM(x non sol) e |b> = 1/sqrt(M)\*SUM(x sol)

**47- Da cosa è composto l’operatore di Grover? Come deriviamo le componenti?**

Scrivo U = HnUHn con P = 2|0X0| - Id (provare per credere). G = UO

Per derivare le componenti basta sviluppare Hn(2|0X0| - Id)Hn

**48- Dai un’interpretazione geometrica e usala per fare le considerazioni**

Lo stato |phi> può essere riscritto come cos(t/2)|a> + sin(t/2)|b>

con cos(t/2) = sqrt((N-M)/N) e sin(t/2) = sqrt(M/N).

Questa riscrittura ci permette di rappresentare lo stato del sistema in uno spazio bidimensionale e legarlo agli angoli, considerando poi i casi per cui N >> M.

Posso allora riscrivere U in termini di seno e coseno e ho che U applicato ad |a>:

(2|phiXphi| - Id) |a> = … = cos(t)|a> + sin(t)|b>

Mentre se applico U a |b> ho: sin(t)|a> - cos(t)|b>

→ da queste due forme derivo U come matrice = ((cos(t), sin(t)), (sin(t), -cos(t)))

→ E dunque G = U \* O = U \* ((1, 0), (0, -1)) = ((cos(t), -sin(t)), (sin(t), cos(t)))

L’operatore di Grover è dunque immediatamente associabile ad una rotazione nel piano definito dagli stati |a> e |b> (non soluzioni e soluzioni). Si vede che se applico G allo stato |delta> ho ((cos(t + delta), sin(t + delta)) cioè l’angolo delta viene ruotato dell’angolo t.

Ne consegue che se applichiamo k volte l’operatore di Grover, genereremo nello spazio una rotazione di un angolo k, e da questo posso derivare le performance scegliendo come angolo delta = t/2 (ottengo cos e sin di ((2k+1)t / 2) → ma approssimo sin((2k+1)t / 2) a 1 → dunque approssimo ((2k+1)t / 2) a pi/2 → quindi k = ½(pi/t - 1) con t = 2 \* sqrt(M/N) perchè approssimo sin(t/2) a t/2 = sqrt(M/N) → inserendo questa relazione in k otteniamo:

k = ½(pi/2 \* sqrt(N/M) -1) =circa= pi/4\*sqrt(N/M) che con N>>M approssimo a sqrt(N).

**Codici correzione errori**

**EXTRA- Come si inizializza un sistema quantistico?**

Vogliamo portare lo stato |0> a un generico cos(a)|0> + sin(a)\*eia|1>: usiamo Uphase(g) che mappa |0> in |0> e |1> in eig|1>. La sequenza è la seguente:

H → Uph(-2\*a) → H → sviluppo, raccolgo |0> e |1> e infine raccolgo e-ia → riscrivo gli esponenziali come 2\*cos(a) e 2i\*sin(a) → raccolgo 2 e semplifico con l’½ della normalizzazione → Uph(- pi/2) che aggiunge -i a i\*sin(a) = i(-i)sin(a) → Uph(g)

**49- Di quali problemi si tratta? Basta il Majority Voting? Quando converge?**

Si tratta di risolvere automaticamente errori di varia natura durante le esecuzioni. Ad esempio, nel caso classico, 000 (0L) può diventare 100. Applicando il Majority Voting (in questo caso 0 vince) avremmo la correzione con ritorno a 0L. Tale operazione converge a 10-4. Nel caso quantistico si ricorda che, per via del no-cloning, è impossibile implentare banalmente la correzione con una copia degli stati 0 e 1.

**50- Come si corregge un Bit Flip?**

Innanzi tutto, dato che la misura potrebbe distruggere lo stato, devo trovare un osservabile in cui io possa misuare senza distruggerlo. Un es. è, per lo stato a|0L> + b|1L>, Z1 (+) Z2.

Misuare con tale osservabile (o con le altre composizioni) mi dice se i due bit sono uguali.

Consideriamo, inoltre, che a|100> + b|001> è anch’esso autostato ma con EW = -1 (e non 1)

Per capire se c’è stato un Bit Flip misuro con Z1 (+) Z2 e poi con Z1 (+) Z3 per individuarlo:

(1, 1) → no err / (1, -1) → err nel terzo bit / (-1, 1) → err nel snd / (-1, -1) → err nel primo

Per correggerlo applico la porta X correttiva al bit di errore precedentemente individuato.

**51- Come si corregge un Weak Error?**

Un weak error, al contrario del bit flip, è un tipo di errore squisitamente quantistico che consiste nella generazione di una sovrapposizione di stati a partire da uno stato singolo (con una probabilità di collasso nel nuovo stato-errore molto più bassa).

|0> →(err): g|0> + d|1> e |1> → -d|0> + g|1> (perchè devono preservare il prod. scalare) con |d| << |g|. Lo stato con l’errore non è autostato di Z1 (+) Z2 e dunque la misura perturberebbe. Dunque lo scriviamo come combinazione lineare di autostati. Raccogliendo:

g(a|000> + b|111>) + d(a|100> - b|011>), che misurando con Z1 (+) Z2 collasserà la maggior parte delle volte nel primo. Nello sfortunato caso in cui collassi nel secondo, posso poi correggere come col bit flip + come col phase.

**52- Come si corregge un Phase Error?**

Notiamo che equivale a un bit flip nella base +/-. Applichiamo dunque 3 porte di Hadamard allo stato e poi consideriamo X1 (+) X2 e X1 (+) X3 esattamente come fatto per il Bit Flip. La correzione, specularmente, avverrà con Z1, Z2 o Z3 a seconda del bit di errore. Si torna poi alla base originaria con la ri-applicazione di Hadamard.

In effetti, si nota che la prima cosa ad essere fatta è l’applicazione di H, e solo poi la “creazione” dell’errore di fase (che a tutti gli effetti è ora un bit flip).

Questo perché un errore di fase nella base 0/1 corrisponde a un bit flip nella base +/-.

**53- Protocollo completo: Come si corregge un Bit Flip?**

Per poter risolvere i 3 errori contemporaneamente abbiamo bisogno di 9 qubit (in realtà è provato che ne bastino 5/7). Definiamo |0L> = (|0’>|0’>|0’>)3 e |1L> corrispondente.

Per il Bit Flip si procede come per il caso a 3 qubit, ma a blocchi (dunque si ha Z1 (+) Z2 e Z1 (+) Z3 per il primo blocco; Z4 (+) Z5 e Z4 (+) Z6 per il secondo; equivalent. per il terzo).

**54- Protocollo completo: Come si corregge un Phase Error?**

Che sia sul primo, secondo o terzo qubit del singolo blocco non ci interessa (perchè cambia il segno del blocco intero, cambiando |0’> in |1’> e viceversa). Verifichiamo con l’eq. agli EW che |0’> sia autostato di “X1 (+) X2 (+) X3” con EW = 1, mentre con |1’> ho EW = -1. Riesco dunque a capire lo stato senza perturbare, misurando (X1 (+) X2 (+) X3)(X4 (+) X5 (+) X6), che possiamo riscrivere come X’1 (+) X’(2). Analogamente al caso a 3 qubit, solo a blocchi, andremo poi a misurare X’1 (+) X’3, e poi a correggere con uno Zj unitario. Si nota che j può essere uno qualsiasi tra i 3 qubit del blocco con l’errore di fase rilevato, dato che qualsiasi flip di bit interno al blocco cambierà la fase.

**55- Protocollo completo: Come si corregge un Weak Error?**

Prendiamo ad es. il primo blocco e generiamo l’errore: |0’> → 1/sqrt(2)\*((g|0> + d|1>)|00> + (g|1> - d|0>)|11>) = … = 1/sqrt(2)\*(g(|000> + |111>) + d(|100> - |011>)). Misurando ora con

Z1 (+) Z2 otteniamo la maggior parte delle volte (con p = |g|2 e EW = 1) lo stato corretto |0>. Le volte in cui otteniamo lo stato con l’errore (ce ne accorgiamo da EW = -1) misuro allora anche con Z1 (+) Z3 per sapere dov’è l’errore (come facevamo col Bit Flip), per poi correggerlo con la porta X corrispondente. Tuttavia, come si nota dallo stato sopra, ad essere errata è anche la fase, e dunque si procede con la correzione del Phase Error come sopra (cioè con X’1 (+) X’2, … e successiva applicazione della porta Z corretta).

Si ripete poi la stessa pipeline anche per gli altri blocchi.

**Quantum Games**

**56- Esponi lo Spin-Flip**

Dati due giocatori P e Q e due stati |0> e |1>:

1) init P a |1> 2) Q decide se applicare X o ID 3) P fa lo stesso 4) Q rifà lo stesso

→ se alla fine lo stato è |1> ha vinto Q, e viceversa. Dato che x2 = Id, se ho applicato X un numero pari di volte sono rimasto sullo stato |1> e dunque ha vinto Q. Viceversa col dispari.

Non esiste dunque una strategia vincente (NB: P e Q non conoscono la strategia dell’altro).

Nel caso quantistico è dimostrabile che esiste una strategia vincente al 100%:

1) init P a |1> 2) Q applica U1 unitaria 3) P applica X con prob = p o Id 4) Q applica U2

Strategia per Q: |1> → a|0> + b|1> → P applica X o Id → Q applica Hadamard, ottenendo a|-> + b|+> nel caso P avesse applicato X, o a|+> + b|-> in caso di applicazione dell’Id da parte di P. Fattorizzando otteniamo 1/sqrt(2)[(a + b)|0> + (b - a)|1>] nel primo caso e uno stato simile nel secondo caso, accomunati dal fatto che scegliendo a = -b si ottenga lo stato |1> con proabilità 1 (sse a = 1/sqrt(2) e b = -1/sqrt(2) → da questo si evince che anche la prima U1 applicata da Q doveva essere per forza di cose Hadamard). Q ha dunque trovato l’autostato di entrambe le porte applicate da P, che dunque non cambierà lo stato qualsiasi cosa egli faccia (|1> →(H): |-> →(ID o X): +/- |-> (aka |-> con fase globale) →(H): |1>).

**57- Esponi il dilemma del prigioniero in generale**

Abbiamo A e B imprigionati che devono scegliere se cooperare (C) o accusarsi (D), con la seguente matrice dei payoff simmetrica per entrambi:

C D Con strategia vincente D, perché vinco 5 o 1 (VS 3 o 0)

C (3,3) (0,5) → si ha dunque un punto di equilibrio di Nesh

D (5,0) (1,1)

Se init a |CC> si applicherà UA (+) UB con U = X o ID (swap C/D o lascia invariato).

Scriviamo Pa = |DDXDD| + 5|DCXDC| + 3|CCXCC| (per Pb avremo 5 su |CDXCD|).

**58- Esponi il dilemma del prigioniero nel caso quantistico**

Sfruttando appieno il caso quantistico (prima potevamo sfruttare solo i casi classici), consideriamo operatori generici: Ua(x, y) = ((eiy\*cos(x/2), sin(x/2)),(-sin(x/2), e-iy\*cos(x/2))

Se x = 0 e y = pi/2 abbiamo Q = ((i, 0), (0, -i)). Data la matrice di init 4x4 Uab, con g = 0 abbiamo l’Identità. Si può dunque, in tal caso, dimostrare che, dato che partiamo stato |CC>, non si ha alcun miglioramento nella strategia rispetto al caso classico (si può anche vedere dal cubo). Se invece poniamo gamma = pi/2 otteniamo uno stato massimamente entangled (anche se diverso dagli stati di Bell) che genera 1/sqrt(2)/(|CC> + i|DD>). In quest’ultimo caso si può dimostrare che la strategia migliore è Q per entrambi con punto di equilibrio di Nesh a (Q, Q) che da un payoff di (3, 3) con miglioramento \*3 rispetto al caso classico.

**59- Esponi il gioco di X/Y in generale**

A, B e C sono stati catturati e ora devono rispondere a una domanda tra queste quattro:

1) {Xa, Xb, Xc}, 2) {Ya, Xb, Yc} 3) {Xa, Yb, Yc} 4) {Ya, Yb, Xc}

Vengono liberati se, dopo la moltiplicazione tra i valori scelti vale il vincolo (-1 per la prima, 1 per gli altri). Si può dimostrare che !esiste una strategia vincente: 1)\*2)\*3)\*4) != 1\*1\*1\*(-1)

**60- Esponi il gioco di X/Y nel caso quantistico**

Introduciamo |GHZ> = 1/sqrt(2)(|000> - |111>) per sfruttare la correlazione e avere una strategia vincente. Se la domanda è Xj misurerò s(x) con collasso in X e EW = +- 1 (e viceversa Yj). Ho dunque che sia X sia Y sono autostati e otterrò EW = +-1 con p=½

La decisione su cosa misurare (X o Y) è condivisa da A e B, mentre obv il risultato dipende dalla misura. Se ad es. voglio chiedere 1) misuro s(x,a) (+) s(x,b) (+) s(x,c) che è diagonale in B2 (in realtà misuro sequenz.). Ho dunque |GHZ> = ¼[(|+>+|->)3 - (|+>-|->)3] = … =

¼[2|++-> + 2|+-+> + …] = ½[|++-> + |+-+> + |-++> + |--->] → misurando avrò in ognuno dei tre casi (con p = ¼) tre EW che moltiplicati danno (-1) come richiesto dal vincolo. Ad es se il sistema collassa in |++-> avrò Xa = 1 dalla prima misura (la p era ¼), poi avrò Xb = 1 con p = 1 dalla seconda misura (è già autostato dunque la misura non perturba= e Xc = -1 con p = 1 dalla terza → prodotto: -1.

Se chiedo 2), 3) o 4) l’unica differenze è la fase (-i)2 generata da |1> = -i(|+’> - |-’>)/sqrt(2),

che cambia il segno ai termini facendo si che a cancellarsi siano i termini con i “-” dispari. Le moltiplicazioni saranno dunque tutte uguali a 1 come richiesto dal vincolo.

**61- Cosa’è la proprietà di non località? Come lo dimostro**

C’è un esperimento che mostra se in natura abbiamo una correlazione classica (palline colorate) o una non località quantistica. Dato che |GHZ> è una combinazione lineare degenere (composto da autostati con stesso EW = 1 per gli operatori del tipo s(x,a) (+) s(y,b) (+) s(y,c)) impone il vincolo di = 1 per le eq. 2), 3) e 4). [v.note]. La misura dell’osservabile s(x,a) (+) s(x,b) (+) s(x,c) determina quale delle due interpretazioni sia quella vera. Basta la misura XaXbXc = -1 per escludere l’interpretazione classica, dato che matematicamente (moltiplicando le 3 equazioni con le Y) si avrebbe il vincolo contraddittorio XaXbXc = 1.

Questa apparente contraddizione deriva dall’aver supposto che si possano attribuire agli stati dei valori fissati per le variabili Xi. Nella interpretazione quantistica, i valori di Xi non sono determinati a priori ma durante la misura.

**Bomb Tester**

**62- Spiega la logica del bomb tester senza assorbente**

Dato un interferometro composto da BeamSplitter (H), specchi M (X) e 2 detectors

D2 → la luce arriva a uno dei due detector

|

M / —----BS - D1 → in questo caso tutta la luce arriverà a D1

| |

| | → con p = ½ il BS redireziona verso l’alto ( |1> )

f (|0>): BS —-----M → con p = ½ il BS redireziona in orizzontale ( |0> )

Ho dunque |0> →BS→ |+> →M→ |+> → BS→ |0> aka D1 con p = 1

**63- Spiega la logica del bomb tester con assorbente**

Se pongo un elemento assorbente (metafora con la bomba) nella direzione verticale dopo il primo BS, ho che gli |1> vengono trasformati in un terzo stato |a>. Questa interferenza si mostra che fa arrivare la luce ad ambo i detector (assumiamo normalizzati):

|0> →BS→ |+> →A→ |0>+|a> →M→ |1>+|a> → BS→ ½(|0>-|1>) + |a>/sqrt(2)

Ho dunque p= ½ che i fotoni vengano assorbiti, ¼ che vengano misurati da D1 e ¼ da D2

Il caso interessante è la misurazione di D2 perchè ci dice che è presente l’elemento assorbente (discriminante rispetto al caso senza di esso).

**64- Spiega la versione migliorata del bomb tester**

Nonostante il 25% delle volte si riesca a sapere se la bomba è presente senza farla esplodere, il 50% di esplosioni è ancora inaccettabile. Si dimostra che si può fare meglio prendendo dei BS non 50/50 ma con prob. tarabile x. Non avrò dunque più BS=H ma avrò BS = U generica (rotazione) = ((cos(x), -sin(x)), (sin(x), cos(x)). Innanzi tutto bisogna, al posto dei detector, collegare dopo il secondo BS un nuovo circuito uguale (o redirezionare i cavi verso il circuito precedente) e ignorare gli specchi. Nel caso con la bomba inattiva:

Se applico N volte U ottengo cos(Nx)|0> + sin(Nx)|1> → scegliendo x = pi/(2N) ho che si annulla il cos e rimane solo |1> → misuro solo il D2 verticale.

Nel caso con la bomba attiva ho che l’assorbimento mi manda ogni volta gli |1> in |a>:

|0> → U → A → cos(x)|0> + sin(x) |a> → U → cos(x)(cos(x)|0> + sin(x) |1>) + sin(x) |a>

→ A → cos2(x)|0> + sin(x)cos(x)|a> + sin(x)|a> → … → cosN(x)|0> + (...)|a>

Ricordandoci che x = pi/(2N) ottengo: [cos(x)]2N c.a= (1-(pi/(2N))2)2N = 1-(pi2/(8N))→(inf): 1

(ricordando che N >> angolo) → ho dunque |0> con probabilità tendente a 1