تصحیح خطای کوانتومی ـ بخش دوم

وحید کریمی پور، دانشکده فیزیک، دانشگاه صنعتی شریف

۱۸ مهر ۱۳۹۸

۱ مقدمه

می دانیم که هرنوع اطلاعات کلاسیک را می توان به صورت رشته ای از علائم 0 و 1 کُد کرد. این نوع کُد کردن اصطلاحاً کُد کردن منبع یا $Source\ Coding$ نامیده می شود و قبل از ارسال اطلاعات به درون کانال انجام می شود و هدف آن تنها این است که اطلاعاتی که به صورت متن، صدا یا تصویر است به صورت رشته ای از علائم ساده و قابل حمل دربیت های کلاسیک درآید. در کُد کردن منبع مسئله‌ای به نام تصحیح خطا وجود ندارد زیرا هنوز اطلاعات به درون کانال که جایی است که خطاها در آن صورت می گیرد، ارسال نشده‌است. به عنوان مثال فرض کنید که منبع ما تنها از چهار حرف A ، B ، C و D تشکیل شده است. یک راه برای کد کردن منبع آن است که این حروف را به رشته های زیر کُد کنیم:

$$\begin{aligned} A &\longrightarrow 00 \\ B &\longrightarrow 01 \\ C &\longrightarrow 10 \\ D &\longrightarrow 11. \end{aligned} \qquad (۱)$$

در مقصد نیز وقتی که این رشته ها به دست گیرنده می رسد او نیز همین روش را برای گشودن کد به کار می برد که به آن $Source\ Decoding$ می گوییم. در حالت ساده بالا این عمل به صورت زیر انجام می شود:

$$00 \longrightarrow A$$

$$01 \longrightarrow B$$
$$10 \longrightarrow C$$
$$11 \longrightarrow D. \qquad (۲)$$

اما وقتی که همین رشته ها را به درون کانال می فرستیم می توانند دچار خطا شوند. به عنوان مثال رشته 01 می تواند دراثر بروز یک خطا در بیت اول تبدیل به 11 شود که در مقصد به صورت حرف D تعبیر خواهد شد. بنابراین برای تصحیح این گونه خطاها که در کانال اتفاق می افتد می بایست در ابتدای کانال یک نوع کد گذاری به کار برد که به آن کدگذاری کانال $ChannelEncoding$ می گوییم . به عنوان مثال یک نوع ساده از کد گذاری کانال آن است که رشته های چهار گانه فوق را به صورت زیر کد کنیم:

$$00 \longrightarrow 0000$$
$$01 \longrightarrow 0101$$
$$10 \longrightarrow 1010$$
$$11 \longrightarrow 1111. \qquad (۳)$$

در گیرنده نیز می بایست این رشته ها را به صورت زیر بگشاییم که به آن $ChannelDecoding$ می گوییم:

$$0000 \longrightarrow 00$$
$$0101 \longrightarrow 01$$
$$1010 \longrightarrow 10$$
$$1111 \longrightarrow 11. \qquad (۴)$$

اینکه چگونه حروف را به صورت رشته های از علایم صفر و یک کد کنیم که هم حداقل بیت ها را به کار ببریم و هم هیچ گونه اطلاعاتی را از دست ندهیم موضوعی است که در فصل های آینده به آن خواهیم پرداخت. در این فصل کار ما این است که تنها امکان بروز خطا و شیوه تصحیح آن را مطالعه کنیم. این که چرا پس از معرفی خطاهای کوانتومی و شیوه تصحیح آنها به بررسی خطاهای کلاسیک و تصحیح خطای کلاسیک می پردازیم، سوالی طبیعی است که ممکن است برای خواننده پیش بیاید. پاسخ اش این است که نظریه کدهای تصحیح خطای کلاسیک به دلیل قدمت آن نظریه ای بسیار غنی است که از جهات گوناگون گسترش یافته است. بخشی از این گسترش با شاخه های کاملا نوینی از ریاضیات نیز تلاقی می کند. علاوه بر این دسته وسیعی از کدهای کوانتومی مبتنی بر ایده هایی هستند که از نظریه کدهای کلاسیک الهام گرفته اند. بعلاوه بر این آشنایی با کدهای کلاسیک ما را با مفاهیم بنیادی ای آشنا می کند که در کدهای کوانتومی نیز معنا و مفهوم دارند. در این درس نخست با

مفاهیم پایه در کدهای کلاسیک آشنا می شویم و سپس به کدهای کوانتومی مبتنی بر آنها می پردازیم. با این مقدمه می توانیم تعریف کلی کدهای کلاسیک را ارائه دهیم. قبل از آن می بایست نماد گذاری خود را روشن کنیم.

■ تعریف: مجموعه $Z_2^n := Z_2 \times Z_2 \times \cdots Z_2$ عبارت است از مجموعه تمام n تایی های مرتب که عناصر آن از 0 و 1 تشکیل شده‌اند.

$$Z_2^n := \{x = x_1x_2x_3\cdots x_n, |x_i = 0, 1\}. \qquad (۵)$$

دقت کنید که در نوشتن عناصر Z_2^n، نمادِ $x_1x_2x_3\cdots x_n$ را بجای نمادِ $(x_1, x_2, x_3, \cdots x_n)$ به کار برده‌ایم. به هر عضو از Z_2^n یک کلمه [۱] می گوییم. تعداد کلمه های Z_2^n برابر است با 2^n .

این مجموعه در ضمن یک فضای برداری روی میدان Z_2 این است یعنی می توان هر کلمه ای را در صفر یا یک ضرب کرد و می توان هر دو کلمه ای را نیز با هم جمع کرد. بنابراین اگر x و y دو عضو از Z_2^n هستند، آنگاه $x + y$ نیز عضوی از Z_2^n است.

■ تعریف : هرگاه $e \in Z_2^n$ یک کلمه‌ی n بیتی باشد، وزن همینگ آن [۲] برابر است با تعداد ۱ های آن. این وزن را با $w(e)$ نشان می دهیم. بنابراین کلمه $e = 001110$ دارای وزن 3 است، یا $w(e) = 3$.

■ تعریف : هرگاه $x, y \in Z_2^n$ دو کلمه‌ی n بیتی باشند، فاصله همینگ آنها برابر است با تعداد بیت هایی که با یک دیگر تفاوت دارند. به عبارت دیگر $d(x, y) := w(x - y)$. این فاصله با $d(x, y)$ نشان داده می شود و می توان نوشت:

$$d(x, y) = \sum_{i=1}^{n} |x_i - y_i| = \sum_{i=1}^{n} (x_i - y_i)^2. \qquad (۶)$$

براحتی می توان نشان داد که فاصله همینگ تمام خصوصیات فاصله را دارد یعنی:

$$d(x, y) \geq 0 \ , \ x = y \longrightarrow d(x, y) = 0 \ , \ d(x, y) = 0 \longrightarrow x = y,$$
$$d(x, y) = d(y, x)$$

[۱] word
[۲] Hamming Weigth

$$d(x,y) \leq d(x,z) + d(z,y). \tag{۷}$$

فرض کنید که کلمه ای مثل v دچار خطای e شود و تبدیل به کلمه $v' = v + e$ شود. دراین صورت تعداد خطاهای صورت گرفته دراین کلمه برابر است با تعداد 1 های درون e که برابراست با $w(e)$. دقت کنید که به ازای هر 1 در e یک خطا روی v ایجاد می شود. به عنوان مثال هرگاه $e = 00110101$ باشد به این معناست که در مکان های ۳ ، ۴ ، ۶ و ۸ کلمه v دچار خطا شده است.

هرگاه احتمال وقوع یک خطا در یک بیت برابر با p باشد و فرض کنیم که خطاهای بیت های مختلف از هم مستقل هستند آنگاه می توان احتمال وقوع یک خطای e را حساب کرد. این خطا برابراست با

$$P(e) = p^{w(e)}(1-p)^{n-w(e)} \approx p^{w(e)}. \tag{۸}$$

که در تساوی تقریبی آخر از این استفاده کرده‌ایم که p معمولا عدد بسیار کوچکی است.

هرگاه $x \in Z_2^n$ یک کلمه‌ی n بیتی باشد، کره هامینگ به شعاع d به مرکز x شامل تمام نقاطی است که فاصله آنها از x مساوی یا کمتراز d است. این کره را با $B_d(x)$ نشان می دهیم:

$$B_d(x) := \{y \in Z_2^n | d(x,y) \leq d\}. \tag{۹}$$

تعداد اعضای درون این کره یعنی تعداد کلمه هایی که در $1,\ 2,\ \cdots$ یا d بیت با x اختلاف دارند. بنابراین تعداد اعضای درون این کره برابر است با:

$$|B_d(x)| = \sum_{i=0}^{d} \begin{pmatrix} n \\ i \end{pmatrix} = 1 + n + \frac{n(n-1)}{2} + \cdots + \begin{pmatrix} n \\ d \end{pmatrix}. \tag{۱۰}$$

ایده اصلی کد گذاری کانال آن است که از 2^n نقطه در فضای Z_2^n تعداد کمتری نقطه انتخاب کنیم و آنها را برای کد کردن 2^k $(k < n)$ کلمه بکار ببریم و این کار را به نحوی انجام دهیم که فاصله این کلمه ها با یک دیگر زیاد باشد به نحوی که اشتباهاتی که در طول کانال رخ می دهد آنها را تبدیل به یک دیگر نکند. بنابراین تعریف رسمی یک کد به صورت زیر است.

■ تعریف : یک کد عبارت است از یک زیر مجموعه از Z_2^n. این زیر مجموعه را با C نشان می دهیم و تعداد اعضای آن را برابر با 2^k می گیریم. به هر عضو C یک کد_کلمه یا $Codeword$ می گوییم. بنابراین می گوییم که کد_کلمه های k C بیت حرف را کد می کنند،

چون تعدادشان برابر با 2^k است.

- تعریف : فاصله یک کدِ C که آن را با $d(C)$ یا d نشان می دهیم برابر است با کمترین فاصله ای که بین کلمات آن وجود دارد. به عبارت دیگر

$$d(C) := min_{x,y \in C} d(x,y). \quad (۱۱)$$

نمادگذاری: کدی را که در فضای Z_2^n نوشته می شود وبرای کد کردن k بیت به کار می رود، و فاصله آن برابر با d است با نمادِ $[n,k,d]$ نشان می دهند.

یک نحوه تصحیح خطاها در یک کد کلاسیک در شکل 1 نشان داده شده است. هرگاه فاصله کد برابر با $d = 2t+1$ باشد، حول هر کدـ کلمه یک کره هامینگ به شعاع t رسم می کنیم. دراین صورت هرگاه نقطه ای دریافت کنیم که متعلق به C نباشد به این معناست که خطایی صورت گرفته است و آن را به صورت کلمه‌ای که نزدیک ترین فاصله را با آن دارد تصحیح می کنیم. به عنوان مثال در شکل 1 کلمه های دریافتی a' و c' متعلق به C نیستند و به کلمه های a و c تصحیح می شوند. به این ترتیب می گوییم که کدی که فاصله آن برابر با $d = 2t+1$ است قادر است که خطاهای با وزن t یا اصطلاحاً t خطا را تصحیح کند.

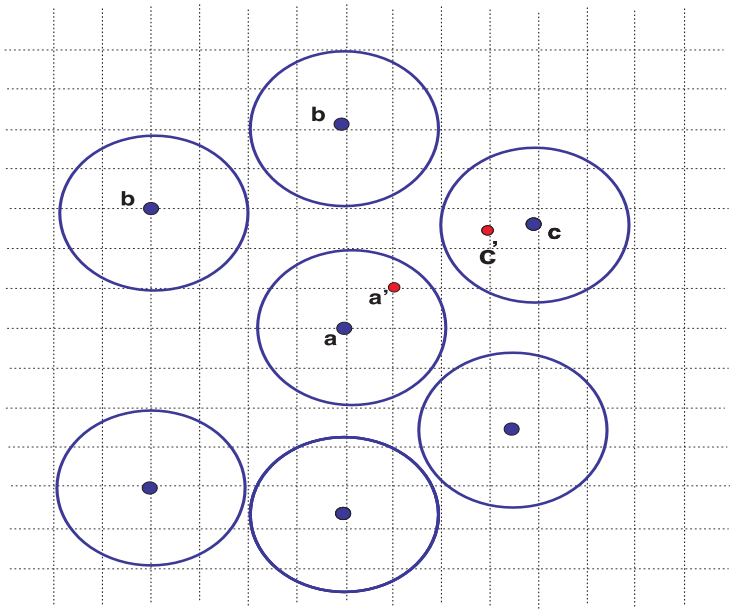
- مثال ۱ : کد زیر را در نظر بگیرید:

$$C = \{000, 111\}. \quad (۱۲)$$

این کد که اصطلاحاً کد تکرار سه تایی نامیده می شود کدی است از نوع $[3,1,3]$ و می تواند یک خطا را تشخیص دهد و تصحیح کند.

- مثال ۲ : کد زیر را در نظر بگیرید:

$$C = \{000000, 101010, 010101, 1111111\}. \quad (۱۳)$$



شکل ۱: نحوه تشخیص و خنثی کردن خطا. اگر کلمه a' را دریافت کنیم آن را به a برمی گردانیم زیرا با احتمال بسیار زیاد این کلمه a بوده است که به a' تبدیل شده است.

این کد از نوع $[6, 2, 3]$ و می تواند یک خطا را تشخیص دهد و تصحیح کند. می توان این کد را برای کد کردن دو بیت به صورت زیر به کار برد:

$$\begin{aligned} 00 &\longrightarrow 000000 \\ 01 &\longrightarrow 010101 \\ 10 &\longrightarrow 101010 \\ 11 &\longrightarrow 111111. \end{aligned} \qquad (۱۴)$$

■ تمرین: آیا می توانید کدی از نوع $[5, 2, 3]$ بنویسید؟

■ یادآوری: توجه کنید که اگر چه خود مجموعه Z_2^n یک فضای برداری است، اما در حالت کلی یک کد تنها یک زیر مجموعه از این فضاست نه یک زیر فضا. در حالت خاصی که کد یک زیرفضاست اصطلاحا می گوییم که یک کد خطی ساخته ایم. این نوع کدها را در بخش های آینده همین درس مطالعه خواهیم کرد.

۲ حدهای حاکم بر کدهای تصحیح کننده خطاهای کلاسیک

برای نوشتن یک کد خوب می بایست بین دو خاصیت متناقض توازن برقرار کنیم. از یک طرف برای آنکه کد خطاهای هرچه بیشتری را تصحیح کند می بایست فاصله آن یعنی پارامتر d بزرگ باشد. به عبارت بهتر می بایست شعاع کره های همینگ ای که در اطراف هر کد کلمه ترسیم می شود زیاد باشد. اما این کار به معنای این است که تعداد کمی کد کلمه را در مجموعه کلمه های n بیتی انتخاب کنیم. به عبارت دیگر نسبت $R := \frac{k}{n}$ که همان نرخ کد است را کم کنیم. بنابراین افزایش فاصله کد با افزایش نرخ کد در تناقض است و یک کد خوب کدی است که این توازن را به بهترین وجهی ایجاد کند. برای این که این محدودیت ها را بفهمیم نخست سعی می کنیم که حد های حاکم بر یک کد دلخواه با مشخاصت $[n, k, d]$ را بفهمیم. این محدودیت ها در سه نامساوی مهم بیان می شوند که در زیر آنها را بیان و ثابت می کنیم. این کد ها را نخست روی بیت ها یعنی کدهایی که روی میدان $Z_2 = \{0, 1\}$ تعریف می شوند مطالعه می کنیم ولی تعمیم آنها به کدهایی که روی الفبای $Z_q = \{0, 1, 2, \cdots q-1\}$ تعریف می شوند نیز سرراست است.

۱.۲ حد همینگ

این حد نخستین بار توسط ریچارد همینگ [۳] مهندس آمریکایی معرفی شده است.

■ قضیه: در هر کد $[n, k, d]$ که روی Z_2 تعریف می شود، نامساوی زیر برقرار است که در آن $d = 2t + 1$:

$$2^k \leq \frac{2^n}{\sum_{i=0}^{t} \binom{n}{i}} \qquad (۱۵)$$

[۳] Richard Hamming

■ اثبات: به دور هر کد کلمه یک کره به شعاع t رسم می کنیم. از آنجا که فاصله کد برابراست با $2d+1$ این کره ها با یکدیگر هیچ اشتراکی ندارند. بنابراین اگر تعداد نقاط درون هر کره را حساب کنیم و در تعداد کره ها ضرب کنیم عدد حاصل می بایست از تعداد کل نقاط کمتر باشد. می دانیم که تعداد کل نقاط برابر است با 2^n. تعداد کل کد کلمه ها برابراست با 2^k . تعداد نقاط درون هر کره به شعاع t را با N_t نشان می دهیم. بنابراین نامساوی همینگ بیان می کند که شرط زیر می بایست برقرار باشد:

$$2^k N_t \leq 2^n. \tag{۱۶}$$

پس کافی است که N_t را محاسبه کنیم. اما محاسبه N_t راحت است و قبلا آن را محاسبه کرده ایم. در واقع :

$$N_t := \sum_{i=0}^{t} \begin{pmatrix} n \\ i \end{pmatrix} \tag{۱۷}$$

زیرا تعداد نقاطی که فاصله آنها برابر با i است برابر با تعداد کلماتی است که دقیقا در i نقطه از n نقطه با کلمه مرکزی تفاوت دارند. با ترکیب این رابطه با رابطه قبلی به حد همینگ می رسیم.

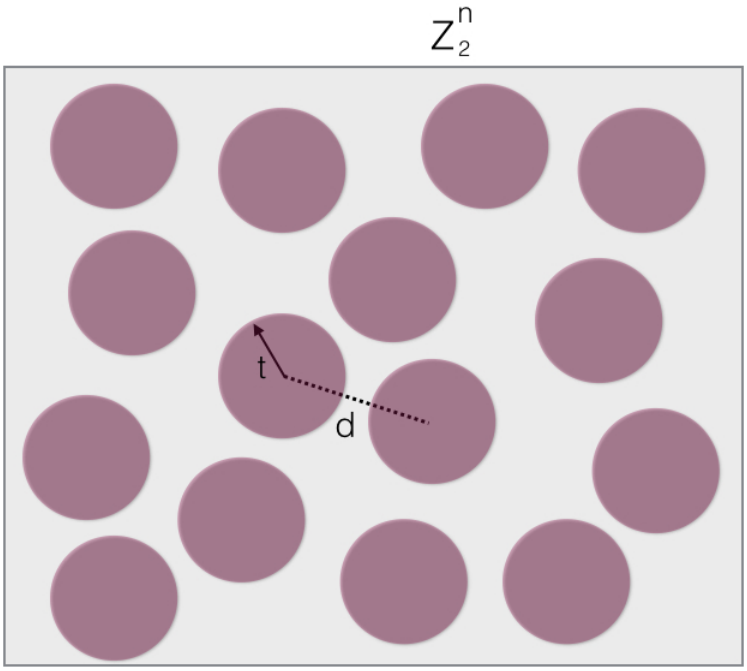
۲.۲ حد گلیبرت ـ وارشاموف

حد همینگ نشان داد که تا چه اندازه می توانیم در فضای Z_2^n کد کلمه ها را بچپانیم. این حد نشان می داد که اگر بخواهیم فاصله کدکلمه ها از یک دیگر زیاد باشد یک حد بالا برای چپاندن این کلمه ها (یا به عبارت بهتر) کره های اطراف انها در فضای Z_2^n وجود دارد. ما همیشه می بایست این حد را رعایت کنیم. مسلم است که با قرار دادن تعداد کمی کره در فضای Z_2^n حد همینگ را رعایت کرده ایم. اما از کجا می توانیم مطمئن باشیم که به صورت بهینه عمل کرده ایم؟ ملاک بهینه بودن یک کد توسط یک نامساوی داده می شود که به آن حد گیلبرت_وارشاموف [۴] می گوییم.

■ قضیه: یک کد $[n,k,d]$ که روی Z_2 تعریف می شود، وقتی بهینه است که نامساوی زیر برقرار باشد :

$$2^n \leq 2^k \times \sum_{i=0}^{d-1} \begin{pmatrix} n \\ i \end{pmatrix}. \tag{۱۸}$$

[۴] Gilbert-Varshamov



شکل ۲: روش بدست آوردن حد همینگ: برای تصحیح t خطا کره های با شعاع t نمی بایست با یکدیگر تلاقی کنند. فاصله کد برابر است با $d = 2t + 1$. برای توضیح بیشتر به متن مراجعه کنید.

به عبارت بهتر هر گاه که این نامساوی نقض شود به این این معناست که ما هنوز می توانیم کدکلمه های دیگری را در فضای Z_2^n قرار دهیم و کد بهتری با نرخ بیشتری بدست بیاوریم. این کار را می توانیم تا آنجا ادامه دهیم تا این نامساوی نقض شود.

- اثبات: فرض کنید که n مقدار معینی دارد. می دانیم که برای اینکه یک کد خوب بنویسیم یعنی کدی که فاصله اش زیاد باشد، می بایست تعداد کد کلمه ها را پایین بیاوریم زیرا می خواهیم فاصله کدکلمه ها از هم زیاد باشد. این همان حدی است که توسط نامسای همینگ داده می شود که بر مبنای آن k نمی بایست از یک حدی بیشتر باشد. اما ما نمی خواهیم که k را خیلی کم بگیریم زیرا در این صورت نرخ کد یعنی $R = \frac{k}{n}$ پایین می آید . برای آنکه یک کد بهینه داشته باشیم می بایست به اندازه کافی کد کلمه در فضای Z_2^n اختیار کنیم و فضای خالی بیهوده ای در آن به جای نگذاریم. این حد توسط گیلبرت ـ وارشاموف [۵] داده می شود. از برهان خلف استفاده می کنیم یعنی

[۵] Gilbert-Varshamov

فرض می کنیم که این نامساوی نقض شود و داشته باشیم

$$\sum_{i=0}^{d-1} \begin{pmatrix} n \\ i \end{pmatrix} \times 2^k < 2^n. \qquad (۱۹)$$

معنای این رابطه این است که کره هایی دور هر کد کلمه به شعاع $d-1$ کشیده شده و مجموع تمام نقاط درون این کره ها بازهم از تمام نقاط درون Z_2^n کمتر است. (دقت کنید که از آنجا که فاصله کد برابر با d است معلوم است که خیلی از کره ها با هم تلاقی پیدا می کنند.) اما از آنجا که بنابر فرض برهان خلف ، کلمه هایی وجود دارند که در درون هیچ کدام از کره ها جای نمی گیرند، پس نتیجه می گیریم که کلمه ای می توان یافت که فاصله اش از همه کلمات دیگر از $d-1$ بیشتر است یا اینکه کلمه ای وجود دارد که فاصله اش از همه کلمات دیگر بزرگتر یا مساوی با d است. پس این کلمه را نیز می توان به کد اضافه کرد بدون اینکه فاصله کد کم شود. یعنی کدی که نامساوی گیلبرت ـ وارشاموف را نقض کند نمی تواند یک کد بهینه باشد. به بیان دیگر ما همواره می توانیم آنقدر کلمه به کد اضافه کنیم تا جایی که این نامساوی نقض شود. [۶]

۳.۲ حد منفرد

- قضیه: در هر کد $[n,k,d]$ نامساوی زیر برقرار است:

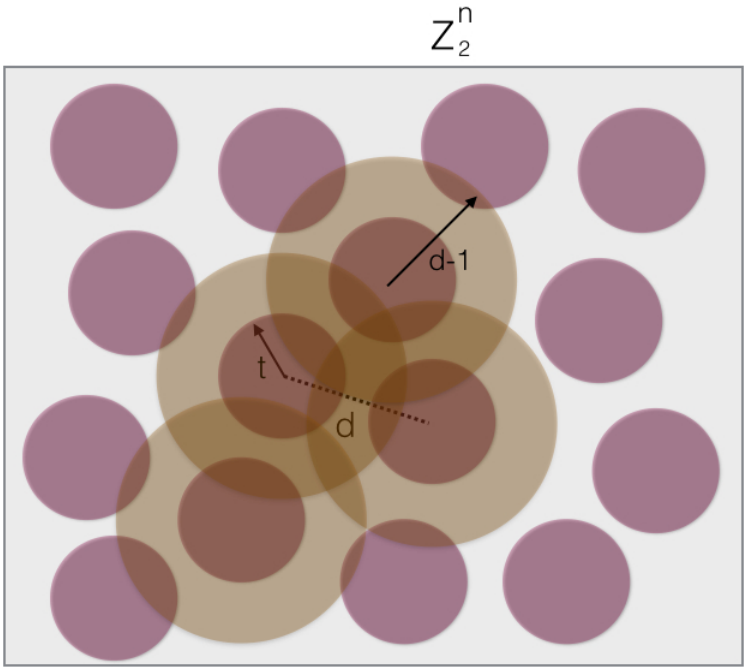
$$k \leq n - d + 1. \qquad (۲۰)$$

این نامساوی که به حد منفرد [۷] مشهور است به سادگی قید روی فاصله، و نرخ کد را مشخص می کند. با وجود اینکه صورت این نامساوی خیلی ساده است ولی اثبات آن نسبت به نامساوی های قبلی احتیاج به فکر و دقت بیشتری دارد.

- اثبات: یک کد کلمه دلخواه مثل $w = (s_1, s_2, \cdots s_{d-1}, s_d, s_{d+1}, \cdots s_n)$ را در نظر بگیرید. تعداد این کد کلمه ها برابر است با 2^k . همه کد کلمه ها چنین ساختاری دارند و فاصله هر دوتای آنها از d بیشتر یا مساوی با آن است. به بیان دیگر هر دو کد کلمه در بیشتر یا مساوی با d حرف با هم تفاوت دارند. پس اگر $d-1$ تا حرف اول را نیز از همه کدکلمه ها پاک کنیم بازهم کد کلمه ها ی جدید برهم منطبق نمی شوند چون که حداقل در یک حرف با هم متفاوت هستند. کلمه های جدید تعداد $n-d+1$ بیت دارند، زیرا $d-1$

[۶] این بیان از حد گیلبرت ـ وارشاموف را مرهون یک بحث با خانم دکتر آسوده هستم.

[۷] Singleton Bound



شکل ۳: روش بدست آوردن حد گیلبرت وارشاموف : یک کد که دارای فاصله $d = 2t + 1$ است، t خطا را تصحیح می کند. کره های با شعاع t هم چنان با یکدیگر تلاقی نمی کنند، اما کره های با شعاع $d - 1$ می توانند با هم تلاقی کنند. با این وجود در یک کد بهینه هیچ کلمه ای نمی بایست بیرون از این کره های بزرگ باقی بماند. زیرا در غیر این صورت آن کلمه را می توانیم به کد خود اضافه کنیم.

تا از بیت های آنها حذف شده است. بنابراین ما تنها با استفاده کردن از $n - d + 1$ بیت توانسته ایم 2^k کلمه متفاوت درست کنیم. پس معنایش این است که شرط

$$2^k \leq 2^{n-d+1}$$

برقرار است که چیزی نیست جز همان حد منفرد. به این ترتیب اثبات این حد نیز کامل می شود.

■ مثال : کد زیر را در نظر بگیرید:

$$C = \{000, 011, 101, 110\}. \qquad (۲۱)$$

در این کد بیت سوم که به بیت پاریته موسوم است پاریته دو بیت اول را تعیین می کند به این معنا که اگر مجموع این دو بیت برابر با صفر باشد مقدار این بیت برابر با صفر و درغیر این صورت مقدار آن برابر با ۱ است. به عبارت دیگر در هر کلمه این کد داریم $x_3 = x_1 + x_2$. با این حساب خواهیم داشت $x_1 + x_2 + x_3 = 0$، یعنی پاریته مجموع سه عدد برابر با صفر خواهد بود. به این تر تیب این کد می تواند یک خطا را آشکار کند، زیرا وقوع یک خطا پاریته کلمه را تغییر خواهد داد. اما این کد تنها می تواند این خطا را آشکار کند و قادر به تصحیح آن نیست . مثلاً معلوم نیست که کلمه دریافتی 111 کدام یک از سه کلمه ارسالی $110, 101, 011$ بوده است که دچار خطا شده است. کمی دقت نشان می دهد که فاصله این کد برابراست با 2 و به همین دلیل است که نمی تواند تشخیص دهد که یک کلمه معیوب ناشی از ایجاد خطا بر روی کدام کلمه کد بوده است زیرا کره های همینگ به شعاع یک برای سه کلمه فوق همگی نقطه‌ی 111 را در بردارند.

■ **مثال :** کد زیر را در نظر بگیرید:

$$C = \{00000, 01011, 01011, 01110, 10011, 10110, 11000, 11101\}. \qquad (۲۲)$$

این کد از نوع $[5, 3, 2]$ و می تواند یک خطا را تشخیص دهد ولی نمی تواند آن را تصحیح کند. دراین کد رابطه زیر برقرار است:

$$x_4 = x_1 + x_2, \qquad x_5 = x_1 + x_2 + x_3. \qquad (۲۳)$$

بنابراین بیت چهارم پاریته دو بیت اول و بیت پنجم پاریته سه بیت اول را در خود نگاه می دارد. سه بیت اول معمولاً بیت های پیام یا $Message\ Bits$ و دو بیت آخر بیت های پاریته یا $ParityChecks$ نامیده می شوند.

$$\begin{aligned} 00 &\longrightarrow 000000 \\ 01 &\longrightarrow 010101 \\ 10 &\longrightarrow 101010 \\ 11 &\longrightarrow 111111. \end{aligned} \qquad (۲۴)$$

این کد از نوع $[6, 2, 3]$ و می تواند یک خطا را تشخیص دهد و تصحیح کند. می توان این کد را برای کد کردن دو بیت به صورت زیر به کار برد:

$$00 \longrightarrow 000000$$

$$01 \longrightarrow 010101$$
$$10 \longrightarrow 101010$$
$$11 \longrightarrow 111111. \qquad (۲۵)$$

احتمالاً اگر سعی کرده باشید که کدی از نوع [5, 2, 3] نویسید متوجه شده‌اید که نوشتن کد ها چندان آسان نیست و قاعده‌ی سرراستی برای ساختن آنها وجود ندارد. کدهای خطی [۸] دسته‌ای از کد ها هستند که خواص بسیار ساده و جالبی دارند و راه و روشی سیستماتیک برای نوشتن کد ها بدست می دهند. در بخش بعدی این کد ها را معرفی می کنیم.

---

۳ کد های خطی

می دانیم که مجموعه Z_2^n یک ساختار خطی دارد و می توان آن را به عنوان یک فضای برداری روی میدان Z_2 در نظر گرفت. هرگاه $\mathbf{x} = x_1x_2\cdots x_n \in Z_2^n$ و $\mathbf{y} = y_1y_2\cdots y_n \in Z_2^n$ آنگاه جمع این دو بردار به شکل زیر تعریف می شود

$$\mathbf{x} + \mathbf{y} = z_1z_2\cdots z_n, \qquad (۲۶)$$

که درآن

$$z_i = x_i + y_i \qquad mod\ 2.$$

هم چنین به صورت بدیهی می توان هر برداری مثل $\mathbf{x}$ را در عددی مثل $\alpha \in Z_2$ ضرب کرد. همه مفاهیمی که برای فضاهای برداری می شناسیم مثل استقلال خطی بردارها، پایه، بعد و نظایر آن برای این فضا نیز تعریف می شوند، با این تفاوت که بایستی همواره در نظر داشته باشیم که میدان اعداد دراینجا میدان اعداد $Z_2 = \{0, 1\}$ است. دراین فضا می توان یک ضرب داخلی نیز به شکل زیر تعریف کرد:

$$\langle \mathbf{x} \cdot \mathbf{y} \rangle = \sum_{i=1}^{n} x_i y_i. \qquad (۲۷)$$

[۸] Linear codes

تعداد عناصر این فضای خطی محدود و برابر با 2^n است.

در قسمت های قبلی کد را به صورت زیر مجموعه‌ای از Z_2^n تعریف کردیم. برای کد های خطی طبیعی است که از خاصیت خطی بودن فضای Z_2^n استفاده کنیم. به همین دلیل کد خطی را به شکل زیر تعریف می کنیم.

■ تعریف : در فضای خطی Z_2^n یک کد خطی چیزی نیست جز یک زیر فضای C از Z_2^n. خاصیت مهم کدهای خطی این است که اگر $(x_1, x_2, \cdots x_n)$ و $(y_1, y_2, \cdots y_n)$ دو کدـ کلمه باشند، آنگاه مجموع آنها یعنی $(x_1+y_1, x_2+y_2, \cdots x_n+y_n)$ نیز یک کدـ کلمه است. هرگاه این زیر فضا یعنی C k بعدی باشد آن را به شکل زیر برای کد کردن k بیت به کار می بریم. چون کد خطی است طبیعی است که همه کدـ کلمه ها را بر حسب بردارهای پایه فضای کد بنویسیم. فرض کنید که $B = \{\mathbf{v}_1, \mathbf{v}_2, \cdots \mathbf{v}_k\}$ یک پایه برای C و $\alpha = \alpha_1\alpha_2\cdots\alpha_k \in Z_2^k$ یک کلمه k بیتی باشد. دراین صورت این کلمه را به صورت زیر در n بیت کد می کنیم:

$$\alpha \longrightarrow \mathbf{v}(\alpha) = \alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \cdots \alpha_k\mathbf{v}_k. \qquad (۲۸)$$

در چنین مواردی می نویسیم

$$C = \langle \mathbf{v}_1, \mathbf{v}_2, \cdots \mathbf{v}_k \rangle. \qquad (۲۹)$$

بنابراین فضای کد خطی دارای بعد k است و دارای 2^k است.

دقت کنید که بردارهای $\{\mathbf{v}_1,\ \mathbf{v}_2, \cdots \mathbf{v}_k\}$ می بایست مستقل خطی باشند که با توجه به اینکه میدان این فضای برداری $Z_2 = \{0, 1\}$ است، به این معناست که مجموع آنها برابر با صفر نیست. بنابراین برای کد $C = \{0000, 1100, 0011, 1111\}$ نمی توان نوشت: $C = \langle 1100, 0011, 1111 \rangle$ زیرا مجموع سه بردار بالا صفر است و مستقل نیستند، بلکه باید نوشت: $C = \langle 1100, 0011 \rangle$

■ مثال: در فضای Z_2^5 یک کد سه بعدی در نظر بگیرید که با بردارهای پایه $B = \{00111, 11000, 10001\}$ جاروب می شود. دراین صورت کلمه‌ی سه بیتی $\alpha = \alpha_1\alpha_2\alpha_3$ به صورت زیر کد می شود:

$$\alpha \longrightarrow \mathbf{v}(\alpha) = \alpha_1(00111) + \alpha_2(11000) + \alpha_3(10001) = (\alpha_2 + \alpha_3, \alpha_2, \alpha_1, \alpha_1, \alpha_1 + \alpha_3). \qquad (۳۰)$$

بنابراین در این کد خطی کلمات سه بیتی به صورت زیر کد می شوند:

$$000 \longrightarrow 00000$$

$$
\begin{aligned}
001 &\longrightarrow 10001\\
010 &\longrightarrow 11000\\
011 &\longrightarrow 01001\\
100 &\longrightarrow 00111\\
101 &\longrightarrow 10110\\
110 &\longrightarrow 11111\\
111 &\longrightarrow 01110.
\end{aligned}
\tag{۳۱}
$$

نوشتن کدهای خطی بسیار آسان است، کافی است که مجموعه‌ای از بردارهای مستقل از فضای Z_2^n در نظر گرفت و کلمات را به صورت ترکیب های خطی انها کد کرد. اما معلوم نیست که انتخاب ما انتخاب خوبی باشد یعنی این کد بتواند خطاها را تصحیح کند. بنابراین سوالی که با آن مواجه هستیم آن است که چگونه می توان خواص کد های خطی نظیر فاصله را تعیین کرد و این که آیا راه ساده‌ای برای تشخیص و تصحیح خطاها توسط این کدها وجود دارد یا نه؟ در واقع بااستفاده از ساختار خطی کد می بایست بتوانیم برای این سوال ها پاسخ های روشنی بیابیم. برای پاسخ به این سوال ها می بایست ساختار کد های خطی را بهتر بشناسیم. این کاری است که در زیر بخش بعدی انجام می دهیم.

---

۴ ساختار کد های خطی

فرض کنید که یک کد خطی با پایه $B_C = \{\mathbf{v}_1, \mathbf{v}_2, \cdots \mathbf{v}_k\}$ تعریف شده باشد. دراین صورت هر کلمه‌ی α به صورت $\mathbf{v}(\alpha)$ کد می شود که درآن

$$
\mathbf{v}(\alpha) = \alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2 + \cdots \alpha_k\mathbf{v}_k = (\alpha_1, \alpha_2, \cdots \alpha_k)\begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \cdots \\ \mathbf{v}_k \end{pmatrix}. \tag{۳۲}
$$

با تعریف ماتریس

$$G = \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \\ \cdots \\ \mathbf{v}_k \end{pmatrix} \tag{۳۳}$$

که بعدِ $k \times n$ دارد می توانیم رابطه بالا را به شکل فشرده زیر بنویسیم:

$$\mathbf{v}(\alpha) = \alpha G. \tag{۳۴}$$

دقت کنید که ماتریس G یک ماتریس با رتبه‌ی k است زیرا همه سطرهای آن مستقل خطی اند. ماتریس G ماتریس مولد [۹] نام دارد.

مثال: کد خطی C با پایه $B_C = \{1000, 1010, 0101\}$ دارای عناصر زیر است:

$$\{0000, 1000, 1010, 0101, 0010, 1101, 1111, 0111\}. \tag{۳۵}$$

ماتریس مولد این کد عبارت است از:

$$G = \begin{pmatrix} 1000 \\ 1010 \\ 0101 \end{pmatrix}. \tag{۳۶}$$

می دانیم که یک کد خطی C با یک زیر فضا در فضای برداری Z_2^n مشخص می شود، شکل 4. این کد برای نوشتن سه بیت در ۴ بیت به کار می رود. هر کلمه سه بیتی مثل $\alpha = \alpha_1\alpha_2\alpha_3$ به صورت زیر به یک کلمه چهاربیتی کد می شود:

$$\alpha \longrightarrow \mathbf{v}(\alpha) = (\alpha_1, \alpha_2, \alpha_3) \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \alpha_1 + \alpha_2 & \alpha_3 & \alpha_2 & a_3 \end{pmatrix} \tag{۳۷}$$

[۹] Generator Matrix

ماتریس مولد یک ماتریس $k \times n$ است که سطرهایش از هم مستقل است یعنی ماتریس است با رتبه k.

- تمرین: از رابطه $\mathbf{v}(\alpha) = \alpha G$ استفاده کنید و ماتریس مولد کد زیر را پیدا کنید.

$$C = \{(a, b, a+b, b, a) \mid a, b = 0, 1\} \tag{۳۸}$$

ماتریس H را برای این بنویسید.

- تمرین: از رابطه $\mathbf{v}(\alpha) = \alpha G$ استفاده کنید و ماتریس مولد کد زیر را پیدا کنید.

$$C = \{(a, b, a+b, c, a+b+c, b+c) \mid a, b, c = 0, 1\} \tag{۳۹}$$

ماتریس H را برای این کد بدست آورید.

فعلا از خطا و چگونگی تصحیح آن صرف نظر می کنیم و از خود می پرسیم که چگونه می بایست کلمه های اصلی را از کلمه های کدشده بازیابی کنیم. پاسخ این سوال ساده است. برای این کار کافی است که بردارهایی مثل

$$\{\mathbf{u}_1, \mathbf{u}_2, \cdots \mathbf{u}_k\} \tag{۴۰}$$

پیدا کنیم که دارای خاصیت زیر باشند:

$$\langle \mathbf{u}_i, \mathbf{v}_j \rangle = \delta_{ij}. \tag{۴۱}$$

هرگاه این بردارها را در ستون های یک ماتریس مرتب کنیم چیزی که بدست می آید همان معکوس ماتریس G است. یعنی

$$G^{-1} = \begin{pmatrix} \mathbf{u}_1 | & \mathbf{u}_2 | & \cdots & | & \mathbf{u}_k \end{pmatrix}. \tag{۴۲}$$

براحتی می توان تحقیق کرد که

$$GG^{-1} = I_k. \tag{۴۳}$$

در نتیجه هر کد کلمه ای که بدست می آید با اعمال این ماتریس تبدیل به کلمه اولیه می شود:

$$\mathbf{v}(\alpha) = \alpha G \longrightarrow \alpha G G^{-1} = \alpha. \tag{۴۴}$$

تا کنون فهمیده ایم که هر ماتریس با رتبه k یک کد خطی تعریف می کند. اما این کد چه خاصیت هایی دارد؟ در اینجا با سوالات ساده ای مواجه می شویم: با در دست داشتن ماتریس G :

- چگونه خطاها را تشخیص می دهیم؟
- چگونه خطاها را تصحیح می کنیم؟
- فاصله کد را چگونه تعیین می کنیم؟

برای پاسخ به سوال اول می بایست روشی بیابیم که به کمک آن تشخیص دهیم آیا یک بردار از فضای کد خارج شده است یا نه؟ اگر به شکل 4 نگاه کنیم می توانیم یک راه ساده برای آن پیدا کنیم. می توان از یک نشانه خیلی ساده برای این کار استفاده کرد. این نشانه ساده ماتریس پاریته [۱۰] خوانده می شود. مجموعه بردارهای عمود بر C را در نظر بگیریم و آن را با $C^{\perp}$ نشان می دهیم.

■ تمرین: نشان دهید که $C^{\perp}$ یک زیرفضای برداری است. هم چنین نشان دهید که بعد این زیرفضا برابر با $n-k$ است.

یک پایه برای این فضا در نظر می گیریم. این پایه از $n-k$ بردار مستقل تشکیل شده است:

$$B_{C^{\perp}} = \{\mathbf{x}_1,\ \mathbf{x}_2,\ \cdots \mathbf{x}_{n-k}\}. \qquad (۴۵)$$

این بردارها را در یک ماتریس H به صورت زیر می نویسیم:

$$H = \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2, \\ \cdots \\ \mathbf{x}_{n-k} \end{pmatrix}. \qquad (۴۶)$$

به عبارت دیگر ماتریس H مولد زیر فضای $C^{\perp}$ است. حال دقت می کنیم که همه این بردارها بر بردارهای $v_i \in C$ عمود هستند، یعنی

$$\langle \mathbf{v}_i, \mathbf{x}_j \rangle = \mathbf{v}_i \mathbf{x}_j^T = 0 \qquad \forall i,\ j.$$

این رابطه را به صورت فشرده می توانیم به این صورت بنویسیم که

$$GH^T = 0. \qquad (۴۷)$$

[۱۰] Parity Check Matrix

از آنجا که همه بردارهای متعلق به C به صورت $\mathbf{v} = v(\alpha) = \alpha G$ هستند، نتیجه می گیریم که هر بردار متعلق به فضای C در رابطه $\mathbf{v}H^T = 0$ صدق می کند. حال اگر بردار v دچار خطا شود و تبدیل به برداری مثل $\mathbf{w} = \mathbf{v} + \mathbf{e}$ شود، خواهیم داشت:

$$\mathbf{w}H^T = (\mathbf{v} + \mathbf{e})H^T = \mathbf{e}H^T \neq 0. \quad (۴۸)$$

بنابراین صفر نشدن wH^T به معنای آن است که بردارِ w یک بردار متعلق به کد نیست و حتماً خطایی اتفاق افتاده است. ولی چگونه می توانیم نوع خطا را بفهمیم و در جهت تصحیح آن اقدام کنیم.

نکته‌ای که وجود دارد آن است که یک بردار معیوب مثل $\mathbf{w}$ ممکن است به طرق مختلفی تولید شده باشد، بعنوان مثال این بردار حاصل خطای $\mathbf{e}_1$ بر بردار $\mathbf{w}_1$ یا خطای $\mathbf{e}_2$ بر بردار $\mathbf{w}_2$ باشد به نحوی که

$$\mathbf{v} + \mathbf{e} = \mathbf{v}' + \mathbf{e} = \mathbf{w}. \quad (۴۹)$$

از میان خطاهای ممکن می بایست محتمل ترین خطا یعنی خطای با کمترین وزن هامینگ در نظر گرفت و فرض کرد که $w = v + e_0$ که در آن e_0 کمترین وزن ممکن را دارد و درنتیجه می بایست w را به شکل زیر تصحیح کرد:

$$w \longrightarrow w + e_0. \quad (۵۰)$$

می توان مطالب بالا را به زبان دقیق و ریاضی نیز بیان کرد. اگر به شکل 4 نگاه کنیم متوجه می شویم که C یعنی خود کُد، یک زیر فضاست و بقیه صفحات در واقع یکسان با C هستند ولی زیر فضا نیستند زیرا شامل بردار 0 نیستند. برای پیشتر رفتن به تعاریف زیر احتیاج داریم.

- تعریف : فرض کنید که V یک فضای برداری و C یک زیر فضای آن باشد. دراین صورت هردو بردار $\mathbf{w}_1, \mathbf{w}_2 \in V$ را هم ارز می گوییم هرگاه رابطه زیر برقرارباشد:

$$\mathbf{w}_1 - \mathbf{w}_2 \in C. \quad (۵۱)$$

خواننده براحتی می تواند ثابت کند که این رابطه یک رابطه هم ارزی است و بنابراین فضای V (دراینجا Z_2^n) توسط زیرفضای C (که دراینجا همان فضای کدـکلمه‌هاست) به زیرمجموعه هایی که عناصر آنها بایکدیگر هم ارز هستند افراز می شود. هر کلاسِ هم ارز یک $Coset$ نامیده می شود. اگر به شکل 4 نگاه کنیم متوجه می شویم که از نظر هندسی هر هم مجموعه یک صفحه موازی با صفحه‌ی C است.

باید دقت کنیم که شکل 4 تنها تا حدودی به درک شهودی ما کمک می کند و از بعضی جهات می تواند گمراه کننده باشد، زیرا ما دراینجا با یک فضای برداری روی میدان $Z_2 = \{0, 1\}$ سروکار داریم نه یک فضای برداری حقیقی. به همین دلیل تعداد بردارهای کل فضا و هم چنین زیرفضای C و تمام کلاس های هم ارزی محدود است. فرض کنید که بردارهای فضای کد یعنی C را به صورت زیر نمایش دهیم:

$$C = \{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \cdots \mathbf{v}_K\} \qquad (۵۲)$$

که درآن $K = 2^k$ (زیرا فرض کرده ایم که C یک فضای k بعدی است.). دراین صورت به ازای هر برداری مثل $\mathbf{e}_i$ که متعلق به C نباشد یک کلاس هم ارزی داریم که برابر است با:

$$[\mathbf{e}_i] \equiv \mathbf{e}_i + C = \{\mathbf{v}_1 + \mathbf{e}_i, \mathbf{v}_2 + e_i, \mathbf{v}_3 + \mathbf{e}_i, \cdots \mathbf{v}_K + \mathbf{e}_i\}. \qquad (۵۳)$$

به دو نکته باید دقت کرد:

■ نکته اول :

برای تمام عناصر یک کلاس اثر H یکسان است، و برابر است با $\mathbf{e}_i H$. بنابراین تمام این کد_کلمه ها یک نشانه خطا دارند. با دریافت این کلمه ها بلافاصله می فهمیم که اشتباهی رخ داده است. البته این اشتباه می توانسته از راه های مختلفی رخ داده باشد. به طور قاطع نمی توانیم بگوییم که کدام خطا رخ داده است. تنها می توانیم بگوییم که محتمل ترین خطا یعنی خطایی با کمترین وزن هامینگ رخ داده و آن خطا را تصحیح کنیم.

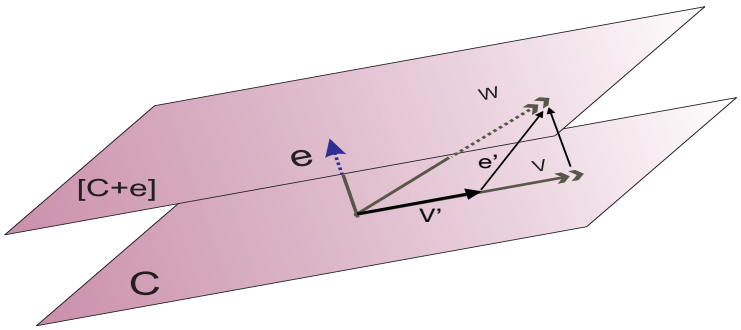
بنابراین اثر H روی یک کد دریافتی تنها به کلاس هم ارزی بستگی دارد و می توان آن را به عنوان شناسنده کلاس هم ارزی یا نشانه خطا[۱۱] بکار برد.

■ نکته دوم :

یک کلاس هم ارزی مثل کلاس 53 را می توان به صورت کلاس $[\mathbf{e}_i + \mathbf{v}_1]$ یا کلاس $[\mathbf{e}_i + \mathbf{v}_2]$ یا کلاس $[\mathbf{e}_i + \mathbf{v}_K]$ نیز نامید و در وهله اول هیچ ترجیحی در نامگذاری یک کلاس به یکی از این صورت ها نیست. از این به بعد نماینده کلاس را برداری می گیریم که کمترین وزن هامینگ را دارد. بنابراین وقتی می گوییم کلاس $[\mathbf{e}_i]$ یعنی بردار $\mathbf{e}_i$ دراین کلاس کمترین وزن هامینگ را دارد.

---

[۱۱] Error Syndrome



شکل ۴: وقتی که کلمه ای مثل w یافته می شود می تواند ناشی از خطای e بر کلمه ی v یا ناشی از خطای e' بر کلمه v' باشد. فرض می کنیم که خطای با کمترین وزن یا کمترین طول همینگ یعنی e رخ داده است و آن را تصحیح می کنیم.

دو نکته فوق به ما می آموزند که چگونه باید خطاها را آشکار کرده و تصحیح کنیم.

نحوه تصحیح خطا: هر بردار $\mathbf{w}$ که دریافت می شود، $\mathbf{w}H^T$ برای آن محاسبه می شود. اگر $\mathbf{w}H^T \neq 0$ می فهمیم که خطایی صورت گرفته است. مقدار $\mathbf{w}H$ در واقع نشانه خطاست و برای ما کلاس هم ارزی $[\mathbf{e}_i]$ را تعیین می کند. دراین کلاس $\mathbf{e}_i$ کمترین وزن هامینگ را دارد و بنابراین محتمل ترین خطا همان $\mathbf{e}_i$ است . بنابراین کلمه دریافتی $\mathbf{w}$ را به صورت $\mathbf{v} = \mathbf{w} + \mathbf{e}_i$ تصحیح می کنیم.

- **خطاهایی که قابل تصحیح نیستند:** ممکن است خطایی که رخ می دهد یک بردار درون صفحه C را به برداری در همان صفحه ببرد. در این صورت چنین خطاهایی هیچ نشانه ای ندارند و قابل تصحیح نیستند. این نوع خطاها یک کد کلمه را به یک کدلمه دیگر تبدیل می کنند.

۱.۴ خواص بیشتری از کد های خطی

تا کنون خواص کد های خطی را بیان کرده ایم ولی هنوز نمی دانیم که چطور یک کد خطی را با خواص معین مثلا فاصله مشخص بسازیم. در این بخش این کار را انجام می دهیم. البته باید توجه کنیم که ساختن کد های خطی خوب، یعنی کدهایی که فاصله زیاد و در عین حال نرخ زیاد داشته باشند، یک کار آسان و بدیهی نیست. یک قضیه مهم به ما کمک می کند که فاصله کد را که یک پارامتر مهم از کد است ، را از روی ماتریس H تعیین کنیم.

۲.۴ فاصله در کدهای خطی

قضیه: هرگاه که هر $d-1$ ستون ماتریس H از هم مستقل خطی باشند، آنگاه فاصله کد برابر است با d.

قبل از اثبات این قضیه بیایید اول نتایج آن را بفهمیم. فرض کنید که می خواهیم کدی بسازیم که یک خطا را آشکار و تصحیح کند. بنابراین فاصله این کد می بایست برابر با 3 باشد. بنابراین قضیه می بایست هر دو ستون ماتریس H از هم مستقل باشند. در فضای برداری ای که روی میدان Z_2 نوشته می شود، دو بردار وقتی مستقل خطی هستند که باهم یکی نباشند. (دقت کنید که این خاصیت فقط برای این نوع فضای برداری درست است.) بنابراین مسئله ما تبدیل می شود به ساختن ماتریسی با ابعاد $n-k \times n$ که هر دوستون اش با هم متفاوت باشند. یک کد خوب می بایست نرخ بالایی نیز داشته باشد به این معنا که k به n نزدیک باشد. بنابراین تعداد سطر های این ماتریس در مقایسه با ستون های آن می بایست هرچه کمتر باشد. این امر به این معناست که می خواهیم تعداد زیادی بردار با مولفه های کم داشته باشیم که همه از هم مستقل باشند و همین تقاضاست که کار ساختن یک کد خوب را دشوار و در عین حال جذاب می کند.

■ مثال:

کد هامینگ را در نظر بگیرید که با ماتریس زیر ساخته می شود:

$$H = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix} \qquad (۵۴)$$

براحتی معلوم می شود که در این ماتریس هر دو ستونی مستقل خطی اند. علاوه بر این می توان سه ستون یافت که وابسته خطی باشند مثل

ستون های ۳، ۴ و ۷. بنابراین فاصله این کد برابر است با 3. در نتیجه این کد با نماد $[7,4,3]$ نمایش داده می شود و دارای نرخ $R=\frac{4}{7}$ است. حال به اثبات قضیه ای که بیان کردیم می پردازیم.

■ **اثبات:** می دانیم که برای هر کلمه متعلق به کُد C شرط $vH^T=0$ برقراراست. بنابراین داریم

$$\sum_i v_i H_{ji}=0 \qquad \forall\, j \tag{۵۵}$$

اما می توانیم این رابطه را به صورت زیر بنویسیم:

$$\sum_i v_i(\mathbf{H}_i)_j=0 \qquad \forall\, j \tag{۵۶}$$

که در آن $\mathbf{H}_i$

بردار یا ستون i ام از ماتریس H است و $(\mathbf{H}_i)_j$ مولفه j ام از این بردار است. از آنجا که رابطه (۵۶) برای همه مولفه ها برقرار است، می توان نوشت:

$$\sum_i v_i\mathbf{H}_i=0. \tag{۵۷}$$

حال توجه می کنیم که بنابرتعریف فاصله یک کُد کمترین مقدار فاصله همینگ بین کلمات آن است. در یک کد خطی فاصله برابر می شود با کمترین وزن همینگ برای خود آن بردارها. در نتیجه هر برداری که از فضای کد در رابطه بالا قرار دهیم حتما تعداد مساوی یا بیشتری از d مولفه برابر با ۱ دارد. یعنی هر d تا ستون ماتریس H به هم وابسته خطی هستند. به همین ترتیب اگر هر $d-1$ تا ستون H از هم مستقل باشند یعنی هیچ کلمه ای با وزن $d-1$ در کد وجود ندارد و در نتیجه کمترین فاصله کد برابر است با d.

به این ترتیب یاد می گیریم که برای ساختن یک کد با مشخصات معین چه مسیری را باید به صورت سیستماتیک طی کنیم. فرض کنید که می خواهیم کدی بسازیم با فاصله 3 که طبیعتا یک خطا را آشکار و تصحیح کند. هم چنین می خواهیم یک بیت را کد کنیم. این به معنای این است که کد ما دارای مشخصات $[n,1,3]$ است. می خواهیم ببینیم با این فاصله یک بیت را می بایست حداقل در چند بیت کد کنیم. چون که $k=1$ است پس می بایست ماتریس H بعد $n-1\times n$ داشته باشد. از آنجا که فاصله کد برابر با $d=3$ است، پس $d-1=2$ است، یعنی هر دو ستون ماتریس H می بایست از هم مستقل یا در واقع با هم متفاوت باشند. پس مسئله ما ساختن ماتریسی $n-1\times n$ است که هر دو ستون آن با هم مساوی باشند. با کمی فکر معلوم می شود که کمترین مقدار n برابر با 3 است و ماتریس H نیز

برابر است با:

$$H = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}. \tag{۵۸}$$

در نتیجه ماتریس G یک ماتریس 1×3 است که می بایست در شرط GH^T صدق کند که تنها جواب اش این است که:

$$G = (1, 1, 1). \tag{۵۹}$$

به این ترتیب هر بیت مثل α در یک سه تایی کد می شود:

$$\alpha \longrightarrow a(1, 1, 1) = (\alpha, \alpha, \alpha). \tag{۶۰}$$

حال اگر بخواهیم دو بیت را در چند بیت کد کنیم که همان فاصله $d = 3$ را داشته باشد باید ماتریس $H_{n-2\times n}$ را چنان پیدا کنیم که هر دو ستون آن از هم مستقل باشند.

- تمرین: کوچکترین ماتریس از نوع بالا را پیدا کنید. سپس ماتریس G و شکل دقیق کدها را بنویسید.
- تمرین: یک کد کلاسیک بنویسید که دارای مشخصات $[n, 2, 3]$ باشد. کوچکترین n را پیدا کنید. ماتریس های G و H را بدست آورید.
- تمرین: یک کد کلاسیک بنویسید که دارای مشخصات $[n, 3, 3]$ باشد. کوچکترین n را پیدا کنید. ماتریس های G و H را بدست آورید.
- تمرین: یک کد کلاسیک بنویسید که دارای مشخصات $[n, 4, 3]$ باشد. کوچکترین n را پیدا کنید. ماتریس های G و H را بدست آورید.

۳.۴ دوگان یک کد

یک کد C دارای ماتریس مولد G و ماتریس چک‌ـپاریته H است که در شرط زیر صدق می کنند:

$$GH^T = 0. \tag{۶۱}$$

بنابراین می نویسیم $C = code(G, H)$ که در آن منظور این است که این کد خطی توسط ماتریس G تولید می شود و توسط ماتریس H چک می شود.

حال اگر طرفین رابطه ۶۱ را ترانهاده کنیم به شرط زیر می رسیم

$$HG^T = 0. \tag{۶۲}$$

این رابطه نشان می دهد که ما می توانیم یک کد دیگر تعریف کنیم که توسط H تولید شده و توسط G چک شود. این کد را دوگان کد C می نامیم و آن را با $C^\perp$ نشان می دهیم. بنابراین می نویسیم

$$C^\perp = code(H, G). \tag{۶۳}$$

دقت کنید که هرگاه یک کلمه متعلق به کد C باشد، داریم:

$$v = \alpha\, G \longrightarrow vH^T = \alpha\, GH^T = 0. \tag{۶۴}$$

برعکس هرگاه یک کد متعلق به کد $C^\perp$ باشد داریم

$$w = \beta H \longrightarrow wG^T = \beta HG^T = 0. \tag{۶۵}$$

هم چنین داریم:

$$\langle w, v\rangle = wv^T = \beta HG^T\alpha = 0 \qquad \forall\, v \in C \qquad , w \in C^\perp. \tag{۶۶}$$

به این ترتیب معلوم می شود که چرا کد دوگان را با $C^\perp$ نشان داده ایم.

- تمرین :نشان دهید که اگر کد C k بیت را در n بیت کد کند ، در این صورت $C^{\perp}$ تعداد $n-k$ بیت را درn بیت کد می کند.

- مثال: کدِ $[5,2,3]$ را در نظر می گیریم. از ما خواسته اند دوگان این کد را بدست آوریم و بگوییم که این کد چه نوع خطاهایی را تشخیص می دهد یا تصحیح می کند.

- حل: نخست توجه می کنیم که این کد می بایست دو بیت را در پنج بیت کد کند. بنابراین ماتریس پاریته آن دارای بعد 3×5 است. چون فاصله کد برابر با ۳ است پس می بایست هر دو سطر این ماتریس از هم مستقل باشند یا در واقع هر دو سطری از این ماتریس با هم متفاوت باشند. به این ترتیب یک انتخاب برای ماتریس H چنین است:

$$H_C = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}. \tag{۶۷}$$

حال ماتریس G به آسانی بدست می آید:

$$G_C = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}. \tag{۶۸}$$

بنابراین بدست می آوریم:

$$G_{C^{\perp}} = H_C = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \tag{۶۹}$$

به این ترتیب این کد سه بیت را در پنج بیت کد می کند. سوال این است که این کد چه خطاهایی را تشخیص می دهد و تصحیح می کند. برای این کار می بایست ماتریس پاریته آن را تشکیل دهیم:

$$H_{C^{\perp}} = G_C == \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}. \tag{۷۰}$$

اما این ماتریس بعضی از سطرهایش با هم یکسان هستند. به این ترتیب فاصله این کد ۳ نیست و از ۳ کمتر است. این کد فقط می تواند خطا در یک بیت را تشخیص دهد ولی نمی تواند آن را تصحیح کند.

- دوگان کد $[7,4,3]$ را بدست آورید و مشخصات آن را معلوم کنید.
- تمرین: نشان دهید که تمام کدـکلمه های $C^{\perp}$ بر تمام کلمه های C عمود هستند. این تمرین نمادگذاری $C^{\perp}$ را برای دوگان یک کد توجیه می کند.

مثال: در Z_2^4 قرار می دهیم:

$$C = \langle 0011, 1100 \rangle = \{0000, 0011, 1100, 1111\} \quad (۷۱)$$

به آسانی معلوم می شود که

$$C^{\perp} = \{0000, 0011, 1100, 1111\} = C. \quad (۷۲)$$

چنین کدی را یک کدِ خودـدوگان می نامیم.

مثال: در Z_2^4 قرار می دهیم:

$$C = \langle 0011 \rangle = \{0000, 0011\}, \quad (۷۳)$$

دراین صورت داریم

$$C^{\perp} = \{0000, 0011, 1100, 1111\} \supset C. \quad (۷۴)$$

۱.۳.۴ کدهای خودـدوگان

هرگاه یک کد دارای این خاصیت باشد که $G = H$ باشد، آنگاه این کد را کد خودـدوگان [۱۲] می نامیم. برای چنین کدهایی داریم: $HH^T = 0$.

برای چنین کدهایی داریم $C = C^{\perp}$. یک نمونه از کدهای خود دوگان کد زیر است:

$$C = \langle 1100, 0011 \rangle = \{0000, 1100, 0011, 1111\} \quad (۷۵)$$

در چنین کدی همه بردارها بر هم، از جمله خودشان، عمودند.

[۱۲] self-dual code

۵ ساختن کدهای کوانتومی از روی کدهای خطی کلاسیک

در ابتدای این درس وقتی که کدهای کلاسیک را مطالعه می کردیم به معرفی کدهای خطی پرداختیم و گفتیم که این کد ها دسته وسیعی از کدهای کلاسیک را تشکیل می دهند. آیا ممکن است که از این کدها ی خطی برای ساختن کدهای کوانتومی استفاده کنیم. این امکان، یک امکان جذاب و درعین حال منطقی است زیرا یک کد کوانتومی بنابر تعریف یک زیرفضای برداری خطی است و این همان چیزی است که در ساختمان کدهای خطی از ابتدا وجود داشته است. در واقع می دانیم که کدهای کلاسیک خطی در واقع به صورت یک زیر فضا از فضای Z_2^n هستند. یعنی اینکه مجموع هر دو کد_کلمه ای خود یک کد_کلمه است. البته این مجموع با ضرایب 0 یا 1 صورت می گیرد. اگر یک کد خطی کلاسیک مثل C داشته باشیم با بردارهای پایه $v_i,\ \ i = 1 \cdots k$ ، براحتی می توانیم یک کد کوانتومی خطی درست کنیم به این تربیت که قرار می دهیم:

$$\mathcal{C}_q := \{|\psi\rangle = \sum_{i=1}^{k} \psi_k |\mathbf{v}_k\rangle, \quad \psi_k = \text{complex number}\}. \tag{۷۶}$$

در واقع کد تکرار سه تایی که یک کوانتومی است دقیقا به همین شیوه از مشابه کلاسیکی آن ساخته شده است. اگر کد کلاسیک $\mathcal{C}$ خطاهای با وزن t را می تواند تصحیح کند، این کد کوانتومی نیز می تواند خطاهای بیت_برگردان با وزن t را تصحیح کند. دلیل این امر هم این است که یک خطای e به این صورت روی کدکلمه های کلاسیک و روی حالت های کوانتومی اثر می کند:

$$\begin{aligned} &Classical: \mathbf{v} = (v_1, v_2, \cdots v_n) \longrightarrow (v_1 + e_1, v_2 + e_2, \cdots v_n + e_n) = \mathbf{v} + \mathbf{e} \\ &Quantum: |\mathbf{v}\rangle = |v_1, v_2, \cdots v_n\rangle \longrightarrow |v_1 + e_1, v_2 + e_2, \cdots v_n + e_n\rangle = |\mathbf{v} + \mathbf{e}\rangle. \end{aligned} \tag{۷۷}$$

حال کافی است که یک عملگر یکانی داشته باشیم که سندروم هر خطایی را تشخیص دهد و این سندروم در حالت کلاسیک همان ماتریس چک_پاریته است و در حالت کوانتومی به شکل عملگریکانی زیر در می آید:

$$U_H : |\mathbf{v}\rangle|0\rangle \longrightarrow |\mathbf{v}\rangle|\mathbf{v}H^T\rangle. \tag{۷۸}$$

■ مثال: در یک کد پنج کیوبیتی ماتریس چک ـ پاریته به صورت زیر است.

$$H = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}. \qquad (۷۹)$$

الف: ماتریس مولد این کد را پیدا کنید.

ب: یک مدار کوانتومی بسازید که حالت های پایه سه کیوبیتی $|\alpha\rangle = |\alpha_1, \alpha_2, \alpha_3\rangle$ را به حالت های پایه پنج کیوبیتی کد کند.

پ: یک مدار کوانتومی بسازید که نشانه های خطا را که با ماتریس پاریته بالا تعریف شده اند تشخیص دهد.

■ حل: الف: هر کلمه ای مثل $\mathbf{v} = (a,\ b,\ c,\ d,\ e)$ که عضوی از کد باشد باید در شرط $\mathbf{v}H^T = 0$ صدق کند. بنابراین چنین کلمه ای حتما می بایست به فرم زیر باشد:

$$\mathbf{v} = (a,\ b,\ a+e,\ b,\ e) \qquad (۸۰)$$

باشد که به این معناست که فضای کد دارای سه بردار پایه است، یعنی

$$\mathbf{v} = a(1,0,1,0,0) + b(0,1,0,1,0) + e(0,0,1,0,1) \qquad (۸۱)$$

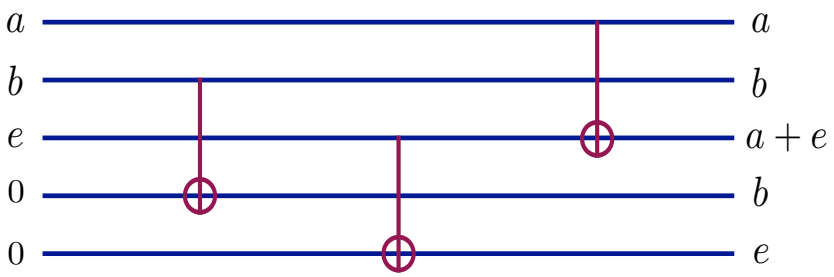
این بردارهای پایه را در یک ماتریس مولد قرار می دهیم و بدست می آوریم:

$$G = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix}. \qquad (۸۲)$$

ب: مدار مولد کد می بایست کار زیر را انجام دهد:

$$|a,b,e\rangle|0,0\rangle \longrightarrow |a,b,a+e,b,e\rangle \qquad (۸۳)$$

براحتی دیده می شود که مدار (۵) این کار را انجام می دهد:



شکل ۵: مدار مولد کد برای ماتریس (۷۹).

پ: با توجه به شکل ماتریس H این مدار باید کار زیر را انجام دهد:

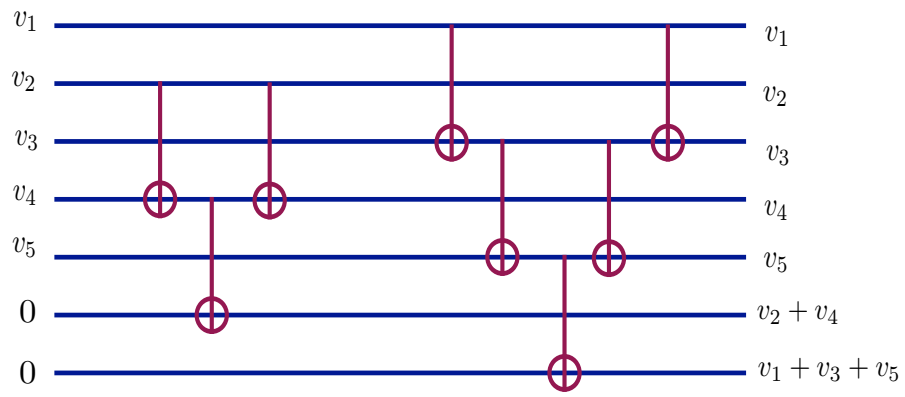
$$|v_1, v_2, v_3, v_4, v_5\rangle|0, 0\rangle \longrightarrow |v_1, v_2, v_3, v_4, v_5\rangle|v_2 + v_4, v_1 + v_3 + v_5\rangle \qquad (۸۴)$$

مدار کوانتومی مربوطه به راحتی ساخته می شود. شکل (۵).

به این ترتیب اگر یک حالت دلخواه مثل $|\psi\rangle$ تحت تاثیر یک خطای بیت ـ برگردان قرار گیرد ، عملگر تشخیص سندروم یعنی U_H می تواند به ترتیب زیر خطا را تشخیص داده و اصلاح کند:

$$\begin{aligned}
&\text{Error :} \qquad X(\mathbf{e}) : |\psi\rangle = \sum_i \psi_i |\mathbf{v}_i\rangle \longrightarrow \sum_i \psi_i |\mathbf{v}_i + \mathbf{e}\rangle, \\
&\text{Error detection} \ : \sum_i \psi_i |\mathbf{v}_i + \mathbf{e}\rangle|0\rangle \longrightarrow \sum_i \psi_i |\mathbf{v}_i + \mathbf{e}\rangle|(\mathbf{v}_i + \mathbf{e})H^T\rangle = \sum_i \psi_i |\mathbf{v}_i + \mathbf{e}\rangle|\mathbf{e}H^T\rangle \\
&\text{Error correction :} \quad X(\mathbf{e}) : \sum_i \psi_i |\mathbf{v}_i + \mathbf{e}\rangle \longrightarrow \sum_i \psi_i |\mathbf{v}_i\rangle = |\psi\rangle. \qquad (۸۵)
\end{aligned}$$

البته کد کوانتومی ای که ساخته ایم تنها می تواند خطاهای بیت‌ـبرگردان را اصلاح کند و توانایی اش برای اصلاح این خطاها به همان اندازه کد



شکل ۶: مدار تشخیص نشانه خطا برای ماتریس (۷۹).

خطی است یعنی خطاهایی با همان وزن را که کد کلاسیک می توانست تصحیح کند این کد نیز می تواند تصحیح کند. چنین کدی را با نماد $[[n, k, d]]$ نشان می دهیم (وجود دو تا علامت براکت به معنای کوانتومی بودن کد است.)

- تمرین: الف: یک کد کلاسیک از نوع $[5, 2, 3]$ در نظر بگیرید. حالت $|\psi\rangle = a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle$ به چه حالتی کد می شود؟

  ب: مدار کوانتومی مولد این کد را بسازید.

- تمرین: الف: یک کد کلاسیک از نوع $[6, 3, 3]$ در نظر بگیرید. حالت های پایه سه کیوبیتی از نوع $(i, j, k) = 0, 1$ $|i, j, k\rangle$ به چه حالت هایی کد می شوند.

  ب: مدار کوانتومی مولد این کد را بسازید.

- تمرین: یک کد کلاسیک از نوع $[7, 4, 3]$ در نظر بگیرید. حالت پایه این کد را وقتی که مطابق بالا آن را تبدیل به یک کد کوانتومی می کنید بدست آورید.

می توانیم پایدارسازهای این کد را نیز بدست آوریم. از آنجا که این کد تنها خطاهای بیت ـ برگردان را تصحیح می کند انتظار داریم که پایدارسازهای آن نیز از عملگر Z ساخته شوند. در واقع از آنجا که به ازای تمام کد کلمه های کلاسیک رابطه $vH^T = 0$ برقرار است می توانیم بنویسیم

$$\forall \ |\mathbf{v}\rangle \in \mathcal{C}_q, \quad (-1)^{\sum_i v_i H_{ji}} |\mathbf{v}\rangle = |\mathbf{v}\rangle, \quad \forall j. \tag{۸۶}$$

و یا

$$\forall \ |\mathbf{v}\rangle \in \mathcal{C}_q, \quad (-1)^{v_1 H_{j1}} (-1)^{v_2 H_{j2}} \cdots (-1)^{v_3 H_{j3}} |\mathbf{v}\rangle = |\mathbf{v}\rangle, \quad \forall j. \tag{۸۷}$$

- با استفاده از رابطه

$$Z|v\rangle = (-1)^v |v\rangle, \qquad v = 0, 1$$

می توانیم رابطه قبلی را به صورت زیر بازنویسی کنیم:

$$Z_1^{H_{j1}} Z_2^{H_{j2}} Z_3^{H_{j3}} \cdots Z_n^{H_{jn}} |\mathbf{v}\rangle = |\mathbf{v}\rangle, \quad j = 1, 2, \cdots n-k. \tag{۸۸}$$

اما حسن بزرگ این نوع نوشتن این است که حالا می توانیم بگوییم که این عملگر هر ترکیب خطی از این بردارها را نیز ثابت نگاه می دارد. به عبارت دیگر این عملگر یک پایدار ساز فضای کد است. به این ترتیب پایدارسازهای این کد را پیدا کرده ایم که برابرند با:

$$S_j := Z_1^{H_{j1}} Z_2^{H_{j2}} Z_3^{H_{j3}} \cdots Z_n^{H_{jn}}, \quad j = 1, 2, \cdots n-k. \tag{۸۹}$$

- مثال: برای کدهای کوانتومی $[[7,4,3]]$ $[[6,3,3]]$, $[[5,2,3]]$, مولدهای پایدارساز را پیدا کنید:

هر کد خطی ای که به این ترتیب می سازیم، تنها می تواند خطاهای بیت‌ـبرگردان را اصلاح کند و از اصلاح خطاهای فازـبرگردان و در نتیجه از اصلاح خطاهای کوانتومی ناتوان است. حال سوال این است که چگونه می توانیم کدی بسازیم که قادر به اصلاح هر دو نوع خطا باشد. این کدها که به طور سیستماتیک با استفاده از دو کد کوانتومی ساخته می شوند به کدهای CSS مشهورند که در بخش آینده به تفصیل آنها را شرح می دهیم.

۶ کدهای CSS

در این بخش به معرفی کدهای CSS [۱۳] می پردازیم. این کدها نخستین بار توسط Calderbank, Steane, Shor معرفی شدند و نام خود را نیز از نام این سه نفر گرفته اند. برای ساختن آنها نیز می بایست دو کد کلاسیک خطی در اختیار داشت. فرض کنید که $C = [n, k, d]$ و $C' = [n, k', d']$ دو کد کلاسیک خطی و $C_q = [[n, k, d]]$ و $C'_q = [[n, k', d']]$ کدهای کوانتومی ای باشند که به طریق ساده بالا از روی آنها ساخته شده اند.

کد C با ماتریس پاریته ی H مشخص می شود که ابعاد آن $(n-k) \times n$ است. به ازای هر سطر این ماتریس یک عملگر پایدارساز به شکل زیر تعریف می کنیم:

$$S_j := Z_1^{H_{j1}} \, Z_2^{H_{j2}} \, Z_3^{H_{j3}} \cdots Z_n^{H_{jn}}. \qquad (۹۰)$$

تعداد این پایدارسازها برابر است با $n-k$. حال به کد C' توجه می کنیم. این کد با ماتریس پاریته H' مشخص می شود که ابعاد آن $(n-k') \times n$ است. به ازای هر سطر این ماتریس نیز یک عملگر پایدارساز به شکل زیر تعریف می کنیم:

$$S'_k := X_1^{H'_{k1}} \, X_2^{H'_{k2}} \, X_3^{H'_{k3}} \cdots X_n^{H'_{kn}}. \qquad (۹۱)$$

طبیعی است که S_j ها بین خود جابجا می شوند و به عنوان سندورم تشخیص خطا ی با وزن $d-1$ به کار می روند. هم چنین S'_k ها بین خود جابجا می شوند و به عنوان سندروم خطای با وزن حداکثر $d'-1$ به کار می روند. ولی آیا می توان این عملگرهای پایدارساز را روی هم ریخت؟ پاسخ در صورتی مثبت است که باهم جابجا شوند. ولی شرط جابجا شدن این عملگرها بسیار ساده است.

■ با استفاده از رابطه جابجایی

$$Z^a X^b = (-1)^{ab} X^b Z^a$$

می توانیم بفهمیم که

$$S_j S'_k = (-1)^{\sum_i H_{ji} H'_{ki}} S'_k S_j, \qquad (۹۲)$$

[۱۳] Calderbank-Steane-Shor

و یا

$$S_j S'_k = (-1)^{(HH'^T)_{j,k}} S'_k S_j. \quad (۹۳)$$

بنابراین پایدارسازهای ۹۰ و ۹۱ وقتی باهم جابجا می شوند که رابطه زیر برقرار باشد:

$$H'H^T = 0. \quad (۹۴)$$

اما این رابطه چه می گوید؟ یک عضو $x \in C'^{\perp}$ را در نظر بگیرید. این عضو توسط H' تولید می شود. بنابراین می توانیم بنویسیم:

$$x = \alpha H' \longrightarrow xH^T = (\alpha H')H^T = 0 \longrightarrow x \in C. \quad (۹۵)$$

بنابراین این رابطه بیان می کند که

$$C'^{\perp} \subset C. \quad (۹۶)$$

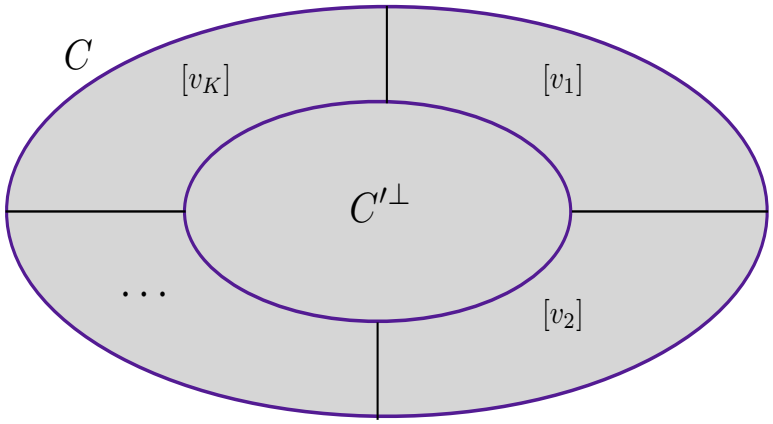
- تمرین: نشان دهید که رابطه زیر نیز برقرار است:

$$C^{\perp} \subset C. \quad (۹۷)$$

- تمرین: با تجربه ای که از حل تمرین های قبلی (برای ساختن کدهای کلاسیک) پیدا کرده اید تحقیق کنید که آیا می توانید دو کد کلاسیک با $n = 5$ پیدا کنید که در شرط $HH'^T = 0$ صدق کنند. با $n = 6$ چطور؟ با $n = 7$ چطور؟ در این مورد پاسخ مثبت است. این کد ها را به طور صریح بنویسید.

پس شرط جابجا شدن تمام عناصر پایدارسازی را که ساختیم پیدا کردیم. تعداد کل این عناصر پایدار ساز برابراست با: $(n-k)+(n-k)$. کد جدید چند تا کیوبیت را در n کیوبیت کد می کند؟ پاسخ برابر است با $n-(2n-k-k) = k+k'-n$. از آنجا که پایدارسازهای S_j تمام خطاهای بیت ـ برگردان تا وزن $d-1$ را شناسایی می کنند و پایدارسازهای S'_k تمام خطاهای فاز ـ برگردان تا وزن $d'-1$ را شناسایی می کنند، معلوم می شود که کد جدیدی که ساخته ایم حتما خطاهای کوانتومی را با وزن d شناسایی می کند که در آن

$$d \geq min(d, d'). \quad (۹۸)$$



شکل ۷: کلاس های هم ارزی $C'^{\perp}$ (زیر فضای افقی) در C. هر کدام از هم مجموعه ها با یک نماینده از آنها مشخص شده است. کلاس $[v]$ شامل تمام کلمه هایی است که با v هم ارز هستند.

بنابراین کد جدید که آن را با نماد $CSS(C,C')$ نشان می دهیم کدی است با مشخصات زیر:

$$CSS(C,C') = [[n, k+k'-n, d \geq min(d,d')]]. \qquad (۹۹)$$

حال سوال می کنیم که حالت های این کد به طور صریح چه هستند؟ برای پاسخ دقت می کنیم که با توجه به رابطه ۹۷ فضای $C'^{\perp}$ زیرفضای C است. در نتیجه رابطه هم ارزی زیر را تعریف می کنیم:

$$\mathbf{v}, \mathbf{v}' \in C, \quad \mathbf{v} \sim \mathbf{v}' \quad if \quad \mathbf{v} - \mathbf{v}' \in C'^{\perp}. \qquad (۱۰۰)$$

این رابطه هم ارزی تمام فضای C را مطابق با شکل ۶ به کلاس ها ی هم ارزی تبدیل می کند. به ازای هر کلاس هم ارزی که یک نماینده مثلv

دارد، یک حالت به صورت زیر تعریف می کنیم:

$$|\overline{\mathbf{v}}\rangle = \sum_{\mathbf{x}\in C'^{\perp}} |\mathbf{v}+\mathbf{x}\rangle. \qquad (۱۰۱)$$

این حالت ها را برای سادگی بهنجار نکرده ایم. در پایان بحث می توانیم با انتخاب ضرایب مناسب آنها را بهنجار کنیم. تعداد کلاس ها برابر است با: $\frac{|C|}{|C'^{\perp}|} = \frac{2^k}{2^{n-k'}} = 2^{k+k'-n}$ که همان مقداری است که انتظار داریم. حال ثابت می کنیم که این حالت ها واقعا توسط عملگرهای پایدارسازی که معرفی کرده ایم پایدار باقی می ماند. نخست می دانیم که به ازای هر بردار $|\mathbf{v}\rangle \in C$ داریم $S_j|\mathbf{v}\rangle = |\mathbf{v}\rangle$ ، در نتیجه واضح است که

$$S_j|\overline{v}\rangle = |\overline{v}\rangle. \qquad (۱۰۲)$$

حال به پایدارسازهای S'_k توجه می کنیم: با توجه به اینکه ماتریس H' مولد کد $C'^{\perp}$ است، داریم

$$S'_k|\overline{\mathbf{v}}\rangle = S'_k \sum_{\mathbf{x}\in C_2^{\perp}} |\mathbf{v}+\mathbf{x}\rangle = X_1^{H'_{k1}} X_2^{H'_{k2}} X_3^{H'_{k3}} \cdots X_n^{H'_{kn}} \sum_{\mathbf{x}\in C_2^{\perp}} |\mathbf{v}+\mathbf{x}\rangle \qquad (۱۰۳)$$

اما می دانیم که ماتریس H' در واقع ماتریس مولد کد $C'^{\perp}$ است که به این معنا ست که سطرهای این ماتریس بردارهای پایه این کد هستند . این بردارهای پایه را با $\mathbf{e}_k$ نشان می دهیم. در نتیجه داریم

$$S'_k|\overline{\mathbf{v}}\rangle = S'_k \sum_{\mathbf{x}\in C'^{\perp}} |\mathbf{v}+\mathbf{x}\rangle = \sum_{\mathbf{x}\in C'^{\perp}} |\mathbf{v}+\mathbf{x}+\mathbf{e}_k\rangle = \sum_{\mathbf{x}'\in C'^{\perp}} |\mathbf{v}+\mathbf{x}'\rangle = |\overline{\mathbf{v}}\rangle. \qquad (۱۰۴)$$

■ تمرین: حالت های ۱۰۱ را بهنجار کنید.

■ تمرین: کد هامینگ یعنی کد کلاسیکِ $[7,4,3]$ را با $C_{Hamming}$ نشان می دهیم . نشان دهید که این کد دارای این خاصیت است که $C^{\perp} \subset C$. چنین کدهایی را کدهای خودـدوگان یا Self Dual می نامند. (راهنمایی: نشان دهید که این کد دارای این خاصیت است که $HH^T = 0$). حال کدِ کوانتومی $CSS(C_{Hamming}, C_{Hamming})$ را بسازید. گروه پایدارساز این کد را تعیین کنید. نشان دهید که واقعا اعضای این گروه با هم جابجا می شوند. مشخصات این کد را بدست بیاورید. بردارهای پایه این کد را بدست بیاورید. به این کد ، کد هفت کیوبیتی [۱۴] می گویند. این کد اگر چه نرخ اش کمتر از کد پنجـکیوبیتی است ولی به دلیل تقارن های خیلی زیادی که دارد، در رایانش مصون از خطا دارای اهمیت است. سندروم های خطاهای یک کیوبیتی را برای این کد بدست آورید و نشان دهید که این کد

[۱۴] 7-qubit code

واقعا تمام خطاهای یک کیوبیتی را اصلاح می کند.