

نظریه اطلاعات کلاسیک

۱ مقدمه

در دروسهای گذشته با مبانی نظری الگوریتم های کوانتومی آشنا شدیم. اما الگوریتم های کوانتومی¹ و محاسبه کوانتومی² تمامی آنچه‌ی نیستند که در این حوزه جدید از دانش وجود دارد. اگر محاسبه کوانتومی و الگوریتم کوانتومی را پردازش داده‌ها و اطلاعات کوانتومی به طرز موثر³ تعریف کنیم آنگاه سوال اساسی که پیش روی ماست آن است که اساساً اطلاعات کوانتومی چیست؟ چگونه می‌توان آن را اندازه گرفت؟ اطلاعات موجود در یک حالت کوانتومی چقدر است؟ تا چه حد می‌توان اطلاعات کوانتومی موجود در یک منبع را فشرده کرد بدون اینکه به محتوای آن صدمه وارد کرد؟ چگونه نوفه‌ی یک کانال کوانتومی موجب از بین رفتن اطلاعات کوانتومی فرستاده شده می‌شود؟ چقدر این اطلاعات از بین می‌رود؟ چگونه می‌توان فهمید که اطلاعات کوانتومی دریافته‌ی آیا نزدیک به اطلاعات ارسال شده هست یا نه؟ چگونه می‌توان به خطاهای ایجاد شده روی اطلاعات کوانتومی پی برد و آنها را آشکار و تصحیح کرد؟ چگونه می‌توان ظرفیت یک کانال کوانتومی را تعریف کرد؟ این سوال‌ها و سوال‌های بسیار دیگر و تلاش برای پاسخ گویی به آنها موضوع اطلاعات کوانتومی را تشکیل می‌دهد. بقیه این درس نامه به این موضوع خواهد پرداخت. اما هر نوع مطالعه‌ای درباره اطلاعات کوانتومی نیازمند آموختن تعاریف و اصول اساسی در باره اطلاعات کلاسیک است. در این درس و درس بعدی با مفاهیم بنیادی اطلاعات کلاسیک آشنا می‌شویم. اگرچه مشابهت های جدی بین نظریه اطلاعات کلاسیک و کوانتومی وجود دارد، ولی بدیهی است که تفاوت های مهم و اساسی بین این دو نظریه نیز وجود دارد. بعد از آشنایی مقدماتی با نظریه اطلاعات کلاسیک به مطالعه نظریه اطلاعات کوانتومی خواهیم پرداخت.

Quantum Algorithm¹
Quantum Computation²
Efficient³

۲ تعاریف اساسی

فرض کنید که $X = \{x_1, x_2, \dots, x_n\}$ یک متغیر تصادفی با احتمالات $\{p_1, p_2, \dots, p_n\}$ باشد. به این متغیر تصادفی می توان تابعی به شکل زیر نسبت داد.

$$H(X) := \sum_{i=1}^n p_i \log_2 \frac{1}{p_i}. \quad (1)$$

بدون اغراق می توان گفت که تمام نظریه اطلاعات کلاسیک بر روی این تابع که آن را تابع آنتروپی می خوانند و خواص و تعبیرهای آن بنا شده است. هدف ما در این درس آن است که اولاً خواص ریاضی این تابع و توابع وابسته به آن را استخراج کنیم، ثانیاً تعبیر و تفسیرهای این توابع را بفهمیم. نخستین کاری که باید بکنیم آن است که نشان دهیم این تابع سنجه مناسبی برای اطلاعات است. این کاری است که در نخستین بخش این درس انجام می دهیم. در بخش های بعدی این درس مفاهیمی مثل اطلاعات شرطی و اطلاعات متقابل را معرفی می کنیم. پس از بررسی خواص ریاضی توابعی که برای اندازه گیری اطلاعات معرفی کرده ایم به فشرده سازی اطلاعات و حدی که برای این فشرده سازی وجود دارد می پردازیم.

۳ مفهوم و اندازه اطلاعات

۱.۳ اطلاعات یک متغیر تصادفی

فرض کنید که آزمایش یا واقعه ای مثل X که نتایج یا پیشامدهای ممکن آن را با مجموعه $\{x_1, x_2, \dots, x_n\}$ نشان می دهیم اتفاق بیفتد و کسی نتیجه این واقعه را به ما بگوید، مثلاً بگوید که پیشامد x_i رخ داده است. در این صورت می توان پرسید که آن شخص چه مقدار به ما اطلاع داده است و چه مقدار از بی اطلاعی ما کاسته شده است. از نظر شهودی هر چقدر که پیشامدی که بوقوع پیوسته است محتمل تر بوده باشد اطلاعی که ما کسب کرده ایم کمتر و هر چقدر که آن پیشامد دور از انتظار بوده باشد تعجب ما از وقوع آن بیشتر و در نتیجه اطلاعی که ما کسب کرده ایم بیشتر خواهد بود. بنابراین اگر میزان اطلاع خود از وقوع پیشامد x_i را با h_i نشان دهیم می توانیم بگوییم که h_i می بایست نسبت معکوس با احتمال وقوع آن پیشامد یعنی p_i داشته باشد. حال فرض کنید که یک آزمایش مرکب از دو واقعه مستقل (X, Y) شود که نتایج ممکن آن را با زوج های $\{(x_i, y_j), i = 1 \dots m, j = 1 \dots n\}$ نشان می دهیم. هرگاه احتمال وقوع x_i را با p_i و احتمال وقوع y_j را با q_j نشان دهیم احتمال هر پیشامد (x_i, y_j) برابر خواهد بود با $p_i q_j$ و میزان اطلاعی که از وقوع این پیشامد کسب می کنیم برابر خواهد بود با $h(p_i q_j)$. انتظار داریم که میزان اطلاع ما در این مورد که دو پیشامد مستقل x_i و y_j رخ داده اند برابر با مجموع اطلاعاتی باشد که از وقوع پیشامد x_i به تنهایی و y_j به تنهایی کسب می کنیم بنابراین انتظار داریم که

$$h(p_i q_j) = h(p_i) + h(q_j). \quad (2)$$

تنهاتابعی که شرط فوق را برآورده کند و ضمناً نزولی باشد، تابع لگاریتم است بنابراین خواهیم داشت:

$$h(p_i) = \log_{\alpha} \frac{1}{p_i}, \quad (3)$$

که در آن α ثابت است. ثابت α را می توان با شرط بهنجارش تعیین کرد. قرار می نهم که میزان اطلاع کسب شده ما از وقوع یک پدیده دو حالتی متساوی الاحتمال برابر یک باشد، یعنی $h(1/2) = 1$. در نتیجه میزان ثابت α برابر می شود با ۲.

اگر یک آزمایش X را N بار انجام دهیم به طور متوسط Np_i بار نتیجه x_i رخ خواهد داد و میزان اطلاع عی که در هر بار کسب می کنیم برابر خواهد بود با $\log_2(\frac{1}{p_i})$. میزان اطلاعی که ما به طور متوسط از وقوع نتایج آزمایش تصادفی X کسب می کنیم برابر خواهد بود با:

$$H(X) = \frac{1}{N} \sum_{i=1}^N Np_i \log_2 \frac{1}{p_i} = \sum_{i=1}^n p_i \log_2 \frac{1}{p_i}. \quad (4)$$

این تابع، تابع آنتروپی یا تابع شانون نیز خوانده می شود. دقت کنید که تابع $p \log \frac{1}{p}$ در فاصله $p \in [0, 1]$ یک تابع مثبت است بنابراین $H(X)$ یک تابع مثبت است.

۲.۳ اطلاعات دو متغیر تصادفی

هرگاه دو متغیر تصادفی (X, Y) داشته باشیم که لزوماً از هم مستقل نباشند تابع آنتروپی یا اطلاعات به طور طبیعی به شکل زیر تعریف می شود:

$$H(X, Y) := - \sum_{i,j} p(x_i, y_j) \log_2 p(x_i, y_j) \quad (5)$$

در حالتی که دو متغیر تصادفی مستقل باشند یعنی $p(x_i, y_j) = p(x_i)q(y_j)$ ، رابطه بالا بدست می دهد که $H(X, Y) = H(X) + H(Y)$.

این تعریف به همین شکل به بیش از دو متغیر تصادفی تعمیم می یابد.

۳.۳ اطلاعات شرطی

دو متغیر تصادفی X, Y که با توزیع آنها با تابع $P(x, y)$ مشخص می شود در نظر می گیریم. فرض کنید که مقدار یکی از متغیرهای تصادفی مثل Y را می دانیم و این مقدار برابر است با y_j . در این صورت توزیع متغیر تصادفی X عوض خواهد شد

و تبدیل خواهد شد به توزیع $P(X|y_j)$ که در آن y_j یک پارامتر است و X مقادیر متغیر را بخود می گیرد. می دانیم که:

$$P(x_i|y_j) := \frac{P(x_i, y_j)}{p(y_j)}, \quad \sum_i p(x_i|y_j) = 1. \quad (6)$$

در نتیجه اطلاعات باقیمانده در متغیر تصادفی X برابر خواهد بود با:

$$H(X|y_j) := - \sum_{x_i} P(x_i|y_j) \log_2 P(x_i|y_j) \quad (7)$$

اگر بخواهیم بدانیم که به طور متوسط دانستن یک مقدار از Y چه مقدار اطلاعات در X باقی می گذارد باید روی $H(X|y_j)$ متوسط بگیریم. بنابراین خواهیم داشت:

$$\begin{aligned} H(X|Y) &= \sum_{y_j} p(y_j) H(X|y_j) = - \sum_{x_i, y_j} P(y_j) P(x_i|y_j) \log_2 P(x_i|y_j) \\ &= - \sum_{x_i, y_j} P(x_i, y_j) \log_2 P(x_i|y_j) = - \sum_{x_i, y_j} P(x_i, y_j) \log_2 \frac{P(x_i, y_j)}{P(y_j)} \\ &= H(X, Y) - H(Y). \end{aligned} \quad (8)$$

دقت کنید که به همان دلیلی که تابع $H(X)$ مثبت است تابع $H(X|y_j)$ و در نتیجه تابع $H(X|Y)$ نیز مثبت خواهند بود. $H(X|Y)$ را اطلاعات X مشروط به Y می خوانیم و این کمیت بیان کننده میزان اطلاعات باقیمانده در X است هرگاه ما مقادیر Y را دانسته باشیم. باید توجه داشت که این تابع متقارن نیست یعنی $H(X|Y) \neq H(Y|X)$.

از رابطه (8) به نتیجه زیر می رسیم:

$$H(X, Y) = H(X|Y) + H(Y) = H(Y|X) + H(X). \quad (9)$$

اگر دو متغیر تصادفی X, Y مستقل باشند آنگاه دانستن Y هیچ تاثیری در اطلاعات باقیمانده در X نخواهد داشت و در نتیجه $H(X|Y) = H(X)$ و بنابر (9)، $H(X, Y) = H(X) + H(Y)$. بالعکس هرگاه X و Y کاملاً به هم وابسته باشند انتظار داریم که دانستن Y برای دانستن X نیز کفایت کند یعنی هیچ اطلاعی در X باقی نگذارد یعنی $H(X|Y) = 0$ که باتوجه به رابطه (9) به این معناست که $H(X, Y) = H(Y)$. این رابطه نیز معنای شهودی روشنی دارد.

۴.۳ اطلاعات متقابل

اطلاعات متقابل در دو متغیر تصادفی X و Y به شکل زیر تعریف می شود:

$$I(X : Y) := H(X) + H(Y) - H(X, Y). \quad (10)$$

این کمیت نسبت به دومتغیرتصادفی X و Y متقارن است. باتوجه به رابطه (9) می توان آن را به شکل زیربازنویسی کرد:

$$I(X : Y) := H(X) - H(X|Y). \quad (11)$$

این رابطه معرف چه چیزی است ؟ قبل ازآنکه مقدار Y را بدانیم، اطلاعات موجود در X با $H(X)$ سنجیده می شد. بآدانستن Y این اطلاعات به $H(X|Y)$ تقلیل پیدا می کند. بنابراین تفاوت این دومیزان اطلاعی است که Y درباره X حمل می کند. بعداً نشان خواهیم داد که $I(X : Y)$ یک کمیت نامنفی است.

۴ یک تذکر مهم درباره اطلاعات منبع

تاکنون اطلاعات منبع X را ازروی فرکانس حروف الفبایی که درآن بکاررفته بود حساب کردیم. بنابراین اگر منبع X را یک متن ادبیات انگلیسی درنظر بگیریم که الفبای آن را بااحتساب نقطه، علامت سوال، فاصله و نظایرآن متشکل از ۳۰ حرف حساب کنیم ازروی فرکانس این سی حرف تابع $H(X)$ را می توانیم محاسبه کنیم. درزبان انگلیسی حرف e بالاترین فرکانس را دارد، پس ازآن حرف t و دیگرحروف قرارگرفته اند. اما بدلیل ساختارمتن ها دوحرف e پشت سرهم یا دوحرف t پشت سرهم بندرت قرارمی گیرند. بنابراین اگر اطلاعات را براساس رشته های یک حرفی و دوحرفی هردودر نظر بگیریم نتیجه متفاوتی بدست می آوریم. درنتیجه راه بهتر و مطمئن تر برای محاسبه اطلاعات منبع آن است که اطلاعات را نه براساس تک تک حروف الفبا بلکه براساس هجاها، کلمات و حتی جملات حساب کنیم. درنتیجه ازاین به بعد وقتی از منبع X و الفبای A صحبت می کنیم منظور ما یک الفبای تعمیم یافته است که عناصر آن نه تک تک حروف بلکه هجاها و کلمات نیز هستند. این پیچیدگی ها درعمل همواره مورد ملاحظه قرارمی گیرند.

۵ خواص ریاضی توابع اطلاعات

دراین بخش خواص ریاضی توابع اطلاعات را بررسی می کنیم. تقریباً همه این خواص از یک قضیه ساده ولی مهم بدست می آیند.

قضیه: فرض کنید که $\{p_i\}$ و $\{q_i\}$ دو توزیع احتمال باشند. دراین صورت

$$\sum_i p_i \log \frac{1}{p_i} \leq \sum_i p_i \log \frac{1}{q_i}, \quad (12)$$

که در آن تساوی فقط وقتی برقراری شود که دو توزیع احتمال یکی باشند. نامساوی بالا را می توان به شکل گویاتر زیرنوشت:

$$\langle \log \frac{1}{p} \rangle_p \leq \langle \log \frac{1}{q} \rangle_p. \quad (13)$$

زیرنویس p بیان می کند که هردومتوسط با توزیع احتمال p محاسبه شده اند.

اثبات: بارسم کردن تابع لگاریتم و تابع $x - 1$ ، می توان نشان داد که تابع لگاریتم در خاصیت زیر صدق می کند:

$$\log x \leq x - 1, \quad (14)$$

که در آن تساوی فقط برای $x = 1$ برقراری شود.

حال قرار می دهیم $x = \frac{q_i}{p_i}$ و در نتیجه

$$\log \frac{q_i}{p_i} \leq \frac{q_i}{p_i} - 1, \quad \forall i, \quad (15)$$

که در آن تساوی فقط وقتی برقراری شود که $q_i = p_i$. در نتیجه

$$\sum_i p_i \log \frac{q_i}{p_i} \leq \sum_i q_i - \sum_i p_i = 0, \quad (16)$$

که همان نامساوی ای است که می خواستیم ثابت کنیم. حال فرض کنید که

$$\sum_i p_i \log \frac{q_i}{p_i} = 0. \quad (17)$$

این تساوی را به شکل زیر بازنویسی می کنیم

$$\sum_i p_i (\log \frac{q_i}{p_i} - (\frac{q_i}{p_i} - 1)) = 0. \quad (18)$$

حال دقت می کنیم که بنابر نامساوی (15) جملات داخل پرانتز همگی کوچک تر از یا مساوی با صفر هستند. صفر شدن این جمع به این معناست که همه این جملات برابر با صفر هستند که با توجه به نامساوی (14) به معنای آن است که برای همه i ها $q_i = p_i$. یعنی دو تابع توزیع احتمال یکی هستند.

نتیجه ۱: مقدار بیشینه تابع اطلاعات $H = \sum_{i=1}^M p_i \log \frac{1}{p_i}$ برابر است با $\log M$ و این مقدار بیشینه را فقط برای توزیع یکنواخت $\{p_i = \frac{1}{M}\}$ اختیار می کند.

اثبات: درقضیه قبلی قراری دهیم $q_i = \frac{1}{M}$. درنتیجه خواهیم داشت:

$$\sum_{i=1}^M p_i \log \frac{1}{p_i} = H - \log M \leq 0, \quad (19)$$

که در آن تساوی فقط وقتی برقراری شود که $p_i = \frac{1}{M}$.

نتیجه ۲: برای دو متغیر تصادفی X, Y نامساوی زیر برقرار است

$$H(X, Y) \leq H(X) + H(Y), \quad (20)$$

که در آن تساوی فقط وقتی برقراری شود که X, Y متغیرهای مستقل باشند.

اثبات: تابع توزیع دو متغیر را با $p(x, y)$ نشان می دهیم. درنتیجه خواهیم داشت:

$$p_1(x) := \sum_y p(x, y), \quad p_2(y) := \sum_x p(x, y). \quad (21)$$

حال تابع توزیع $q(x, y) := p_1(x)p_2(y)$ را در نظر می گیریم و از قضیه ای که ثابت کردیم استفاده می کنیم:

$$\sum_{x, y} p(x, y) \log \frac{q(x, y)}{p(x, y)} \leq 0 \quad (22)$$

که در آن تساوی فقط وقتی برقراری شود که $p(x, y) = q(x, y) = p_1(x)p_2(y)$. اما نامساوی بالا را وقتی باز نویسی کنیم چیزی نیست جز

$$H(X, Y) \leq H(X) + H(Y). \quad (23)$$

این عبارت بیان می کند که به عنوان مثال اطلاعات موجود در جمله « فردا هوا ابری است و باران می بارد » کمتر از مجموع اطلاعاتی است که در دو جمله « فردا هوا ابری است » و « فردا هوا بارانی است » می باشد. دلیل این امر آن است که معمولاً بین ابری بودن هوا و بارانی بودن آن یک همبستگی وجود دارد که به ما اجازه می دهد از اولی بتوانیم وجود دومی را حدس بزنیم. بنابراین کسی که هر دو جمله را به ما می گوید دوبرابر کسی که فقط یکی از جملات را به ما می گوید به ما اطلاع نمی دهد.

نتیجه ۳: اطلاعات متقابل یک کمیت نامنفی است. این نتیجه از تعریف اطلاعات متقابل و نتیجه ۲ بدست می آید.

قضیه: اطلاعات تابع محدبی از توزیع احتمال است. به عبارت دیگر اگر P_1 و P_2 دوتابع توزیع احتمال و $P_0(x) = \lambda P_1(x) + (1 - \lambda)P_2(x)$ ترکیب خطی محدب آنها باشد آنگاه

$$H_0(X) \geq \lambda H_1(X) + (1 - \lambda)H_2(X). \quad (24)$$

به اصطلاح می گوییم که اطلاعات یک تابع محدب روبه پایین است که به یادماندن شکل آن را نیز در ذهن آسان می کند.

اثبات: بازهم از نامساوی اساسی ای که ثابت کردیم استفاده می کنیم. با کمی خلاصه نویسی در نمادها خواهیم داشت:

$$\begin{aligned} & H_0 - \lambda H_1 - (1 - \lambda)H_2 \\ &= \sum p_0 \log \frac{1}{p_0} - \lambda \sum p_1 \log \frac{1}{p_1} - (1 - \lambda) \sum p_2 \log \frac{1}{p_2} \\ &= \sum (\lambda p_1 + (1 - \lambda)p_2) \log \frac{1}{\lambda p_1 + (1 - \lambda)p_2} - \lambda \sum p_1 \log \frac{1}{p_1} - (1 - \lambda) \sum p_2 \log \frac{1}{p_2} \\ &= \lambda \sum p_1 \log \frac{p_1}{\lambda p_1 + (1 - \lambda)p_2} + (1 - \lambda) \sum p_2 \log \frac{p_2}{\lambda p_1 + (1 - \lambda)p_2} \geq 0, \end{aligned} \quad (25)$$

که در خط آخر از نامساوی اساسی استفاده کرده ایم.

تعریف کانال کلاسیک: منظور از یک کانال کلاسیک عملگری است که یک آنزامل تصادفی X را به آنزامل تصادفی Y تبدیل می کند. بهترین مثال آن هرنوع کانال مخابراتی کلاسیک است. X را ورودی کانال و Y را خروجی آن می نامیم. یک کانال بدون نوفه کانالی است که خروجی آن دقیقاً با ورودی آن برابر است. بجزاین کانال ایده آل هرکانال دیگری علائم ورودی $x_i \in X$ را با احتمالات معین $P(y_j|x_i)$ به علائم خروجی $y_j \in Y$ تبدیل می کند. هرگاه در خروجی کانال علامت y_j را دریافت کنیم می توانیم احتمال شرطی این که چه علامت x_i ای منجر به این علامت در خروجی شده است را حساب کنیم. درواقع داریم:

$$\begin{aligned} P(x_i|y_j) &= \frac{P(x_i, y_j)}{P(y_j)} = \frac{P(y_j, x_i)}{\sum_{x_i} P(y_j, x_i)} \\ &= \frac{P(y_j|x_i)P(x_i)}{\sum_{x_i} P(y_j|x_i)P(x_i)} \end{aligned} \quad (26)$$

در آخرین عبارت $P(x_i)$ مشخصه منبع X و $P(y_j|x_i)$ مشخصه کانال است و هر دو معلوم هستند.

قضیه: اطلاعات پردازش شده در یک کانال $I(X; Y)$ تابع محدبی از احتمالات ورودی X است.

اثبات: در یک کانال آنزامل ورودی را با X و آنزامل خروجی را با Y نشان می دهیم. احتمالات شرطی $P(y|x)$ درواقع مشخصه کانال هستند و احتمال تبدیل پیام x به y را در طول کانال نشان می دهند و ربطی به احتمال پیام

های ورودی ندارند. حال هرگاه برای آنزامل ورودی دو تابع توزیع احتمال $P_1(x)$ و $P_2(x)$ و جمع محدب آنها یعنی $P_0(x) = \lambda P_1(x) + (1 - \lambda)P_2(x)$ را در نظر بگیریم آنگاه باتوجه به تعاریف زیر:

$$\begin{aligned} P(y) &= \sum_x P(y|x)P(x), \\ P(x, y) &= P(y|x)P(x), \end{aligned} \quad (27)$$

خواهیم داشت

$$\begin{aligned} P_0(x, y) &= \lambda P_1(x, y) + (1 - \lambda)P_2(x, y) \\ P_0(y) &= \lambda P_1(y) + (1 - \lambda)P_2(y). \end{aligned} \quad (28)$$

باترکیب این روابط با تعریف اطلاعات متقابل و هم چنین محدب بودن تابع اطلاعات اثبات قضیه کامل می شود.

مثال: جفت متغیر تصادفی (X, Y) را مطابق جدول زیر در نظر بگیرید: Y ناشی از انداختن یک طاس است که مقادیر ۱ تا ۶ را به خود می گیرد و X نیز دو مقدار متفاوت یک سکه است که مقادیر a یا b را اختیار می کند.

(X, Y)	1	2	3	4	5	6
a	0.2	0.1	0.08	0.04	0.05	0.05
b	0.1	0.02	0.15	0.06	0.1	0.05

(29)

کمیت های زیر را حساب کنید: الف: $H(X)$ ، $H(Y)$ ، $H(X, Y)$ ، $H(X|Y)$ ، $H(Y|X)$ و $I(X; Y)$.

۶ فشرده سازی اطلاعات در غیاب نوفه

بهترین کاربرای فهم فشرده سازی اطلاعات مطالعه یک مثال ساده است. فرض کنید که هدف ما ارسال متن هایی است که تنها از چهار حرف الفبا به نام های A ، B ، C و D تشکیل شده است. یک روش برای ارسال این متن ها آن است که حرف های چهارگانه فوق را با بیت های 0 و 1 که در مخابرات دیجیتال معمول است، به ترتیب زیر کد کنیم.

$$\begin{aligned} A &\longrightarrow 00 \\ B &\longrightarrow 01 \\ C &\longrightarrow 10 \\ D &\longrightarrow 11. \end{aligned} \quad (30)$$

در این صورت به ازای هر حرف دوبیت مخابره کرده ایم. حال سوال این است که آیا می توانیم یک روش کد کردن به کاربریم که در آن طول به ازای هر حرف تعداد بیت هایی که به طور متوسط مخابره می کنیم کمتر از 2 باشد؟ فرض کنید که این حروف در متن های یادشده با احتمالات زیر ظاهر می شوند:

$$P(A) = \frac{1}{8} \quad P(B) = \frac{1}{8} \quad P(C) = \frac{1}{4} \quad P(D) = \frac{1}{2}. \quad (31)$$

حال روش کدگذاری زیر را به کار می بریم:

$$\begin{aligned} D &\longrightarrow 0 \\ C &\longrightarrow 10 \\ B &\longrightarrow 110 \\ A &\longrightarrow 111. \end{aligned} \quad (32)$$

در این روش کدگذاری برای بعضی از حروف بیش از دوبیت به کار برده ایم ولی اگر طول متوسط کدهایی را که برای حروف به کار برده ایم محاسبه کنیم نتیجه جالب توجه خواهد بود. این طول متوسط برابر است با:

$$\langle l \rangle = \sum_{i=1}^4 l_i \times p_i = 1 \times \frac{1}{2} + 2 \times \frac{1}{4} + 3 \times \frac{1}{8} + 3 \times \frac{1}{8} = \frac{7}{4}. \quad (33)$$

بنابراین با یک کدگذاری مناسب توانسته ایم طول متوسط رشته بیت هایی را که برای مخابره پیام بکار برده ایم از ۲ به ۷/۴ تقلیل دهیم. ضمناً باید دقت کنیم که این نحوه کدگذاری هیچ نوع ابهامی درباره متنی که مخابره شده است دربر ندارد و هر رشته ای از بیت ها به طور یکتا به متن اولیه بازگشایی می شود. به عنوان مثال رشته زیر

$$010001000110111. \quad (34)$$

بدون ابهام به متن زیرگشوده می شود و متن دیگری برای بازگشایی آن قابل تصور نیست

$$D C D D C D D B A. \quad (35)$$

این که چه نوع کد هایی یکتاگشاهستند موضوعی است که مادردهای آینده به آن خواهیم پرداخت و فعلاً موضوع بحث مانیت. ولی یک نکته مهم را باید ذکر کنیم: هرگاه آنتروپی متغیر تصادفی $X = \{A, B, C, D\}$ را با احتمالات ذکر شده حساب کنیم حاصل آن برابر خواهد بود با:

$$H(X) = \sum_{i=1}^4 p_i \log_2 \left(\frac{1}{p_i} \right) = \frac{1}{2} \times \log_2(2) + \frac{1}{4} \times \log_2(4) + \frac{1}{8} \times \log_2(8) + \frac{1}{8} \times \log_2(8)$$

$$= \frac{1}{2} \times 1 + \frac{1}{4} \times 2 + \frac{1}{8} \times 3 + \frac{1}{8} \times 3 = \frac{7}{4}. \quad (36)$$

بنابراین در این مثال خاص طول متوسط کدگذاری ای که به کار بردیم با میزان اطلاعات موجود در متن برابر است. آیا این یک خصلت عمومی است؟

بعد از ذکر این مثال می خواهیم بفهمیم که در حالت کلی چگونه می توان اطلاعات موجود در یک منبع X را فشرده کرد. فرض کنید که منبع متن هایی تولید می کند که این متن ها از الفبای $A = \{x_1, x_2, \dots, x_N\}$ تشکیل شده اند و احتمال ظاهر شدن هر حرف مثل x_i در این متن ها با p_i داده می شود. بنابراین یک منبع رami توان به عنوان یک متغیر تصادفی با اطلاعات معین $H(X)$ در نظر گرفت. برای سادگی فرض کنید که N توانی از 2 است یعنی $N = 2^n$. حال اگر بدون توجه به احتمالات ظاهر شدن حروف مختلف بخواهیم متن ها را مخابره کنیم می توانیم هر حرف الفبای A را بایک رشته n تایی از بیت های 0 و 1 کدگذاری کنیم. در این صورت برای هر متن که شامل M حرف است تعداد Mn بیت مصرف می کنیم یا به عبارت دیگر به ازای هر حرف الفبا n بیت مصرف می کنیم.

ولی می توانیم روش کدگذاری بهتری به ترتیب زیر بکار ببریم. به جای اینکه تک تک حروف الفبا را کدگذاری کنیم، رشته های طولانی به طول m حرف را در نظر می گیریم و برای هر رشته یک کد انتخاب می کنیم. تعداد کل رشته های m حرفی ممکن که از الفبای A تشکیل شده اند برابر است با N^m . ولی نکته در این است که احتمال ظاهر شدن بسیاری از رشته ها آنقدر ناچیز است که نیازی به کد کردن آنها نیست و یا کد کردن تنها رشته های متعارف (رشته هایی که زیاد ظاهر می شوند) چیزی از دست نمی دهیم. به این ترتیب یعنی با کد کردن تنها رشته های متعارف ماقادری شویم که بیت های کمتری برای مخابره متن های منبع X مصرف کنیم. اما رشته های متعارف کدام ها هستند؟ و کد کردن آنها چقدر باعث فشرده شدن پیام هاست؟ در هر رشته m حرفی به شرطی که m به اندازه کافی بزرگ باشد به تقریب تعداد mp_1 حرف آن x_1 ، mp_2 حرف آن x_2 و mp_N تا حرف آن x_N خواهد بود. هر قدر که طول رشته یعنی m بیشتر باشد، افت و خیز تعداد واقعی حرف ها حول این مقادیر متوسط کمتر خواهد بود. حال سوال می کنیم که چه تعداد رشته متعارف وجود دارد. تعداد این رشته های متعارف تقریباً برابر است با تعداد طرقي که می توان از یک رشته m بیتی mp_1 تا mp_N را به صورت x_1 انتخاب کرد و mp_2 تا x_2 و الی آخر. این تعداد برابر است با:

$$K = \frac{m!}{(mp_1)!(mp_2)! \dots (mp_N)!} \quad (37)$$

اما با استفاده از تقریب استرلینگ می توانیم بنویسیم:

$$\log_2 K = \log_2 \left(\frac{m!}{(mp_1)!(mp_2)! \dots (mp_N)!} \right) \approx m \left(\sum_{i=1}^N p_i \log_2 \frac{1}{p_i} \right) \equiv mH(X) \quad (38)$$

که در آن تابع $H(X)$ به صورت زیر تعریف شده است:

$$H(X) := \sum_{i=1}^N p_i \log_2 \left(\frac{1}{p_i} \right) \quad (39)$$

بنابراین تعداد جملات متعارف باتقریب بسیارخوب برابرخواهد بود با

$$K \approx 2^{mH(X)} \quad (40)$$

حال اگر تعداد جملات متعارف برابر باشد با مقدار فوق، می توانیم هرکدام از این جملات را با یک رشته بیت های 0 و 1 کدگذاری کنیم و مسلم است که تعداد بیت هایی که برای این کار احتیاج داریم برابر است با $mH(X)$. از آنجا که هر رشته دارای m حرف بوده است مثل این است که در عمل برای مخابره هر حرف $H(X) := k$ بیت بکار برده ایم. از آنجا که $H(X) \leq \log_2 N = n$ از آنجا که n کمتر است.

Block coding می گوئیم برای هر حرف به جای n بیت $H(X)$ بیت مصرف کرده ایم که از n کمتر است. آنچه که در بالا گفته شد محتوای کلی قضیه شانون در مورد کدگذاری بدون نوبه بود. ولی چگونه می توان این حرف را دقیق کرد؟ چگونه می توان تعریف دقیقی از رشته های متعارف بدست داد؟ با کد نکردن رشته های غیر متعارف چه مقدار مرتکب خطا می شویم؟ آیا بیش از این هم می توان پیام های منبع X را فشرده کرد؟ برای پاسخ به این سوالات سعی می کنیم ابتداتعارف دقیقی از مفاهیم گفته شده بدست دهیم.

۷ تعریف دقیق از رشته های متعارف

تعریف: رشته $\alpha = \alpha_1 \alpha_2 \alpha_3 \dots \alpha_m$ را در نظر بگیرید. تعداد حروف x_j در این رشته را با $f_j(\alpha)$ نشان دهید. تعداد حروف x_j در رشته های به طول m به طور متوسط برابر است با mp_j و واریانس توزیع احتمال حول این مقدار متوسط برابر است با $\sigma_j := \sqrt{mp_j(1-p_j)}$. رشته متعارف رشته ای است که تفاوت تعداد واقعی هر کدام از حروف مثل x_j از تعداد متوسط آن یعنی mp_j در مقایسه با واریانس σ_j کوچک باشد. بنابراین به ازای هر ϵ عدد k را چنان انتخاب می کنیم که $\frac{1}{k^2} < \frac{\epsilon}{N}$. در این صورت رشته α رشته متعارف خوانده می شود اگر شرط زیر برقرار باشد:

$$\left| \frac{f_i(\alpha) - mp_i}{\sqrt{mp_i(1-p_i)}} \right| < k \quad \forall i = 1, 2, \dots, N. \quad (41)$$

تعداد رشته های متعارف برای m های بزرگ در حدود 2^{mH} است که هر کدام با احتمال 2^{-mH} در این مجموعه پدیدار می شوند. به عبارت دقیق تر قضیه زیر برقرار است.

قضیه:

الف: احتمال کل رشته های غیر متعارف از ϵ کمتر است.

ب: یک عدد A وجود دارد به قسمی که به ازای هر رشته متعارف α

$$2^{-mH-A\sqrt{m}} < P(\alpha) < 2^{-mH+A\sqrt{m}}. \quad (42)$$

پ: تعداد رشته های متعارف به طول m برابر است با $2^{m(H+\delta_m)}$ که در آن $\lim_{m \rightarrow \infty} \delta_m = 0$.

اثبات: نخست به یک لم احتیاج داریم.

لم: نامساوی چبیشف (Chebyshev inequality):

الف: فرض کنید که متغیر تصادفی X مقادیر مثبت $\{x_1, x_2, \dots, x_N\}$ را با احتمالات $\{p_1, p_2, \dots, p_N\}$ اختیار می کند. در این صورت به ازای هر عدد مثبت α ,

$$P(X \geq \alpha) \leq \frac{\bar{X}}{\alpha} \quad (43)$$

که در آن $\bar{X}$ متوسط متغیر تصادفی X است.

اثبات:

$$P(X \geq \alpha) = \sum_{x=\alpha}^{\infty} P(x) \leq \sum_{x=\alpha}^{\infty} \frac{x}{\alpha} P(x) \leq \frac{\bar{X}}{\alpha}. \quad (44)$$

ب: حال فرض کنید که متغیر تصادفی X مقادیر دلخواه مثبت یا منفی اختیار می کند. در این صورت به ازای هر عدد k

$$P((X - \bar{X})^2 \geq k^2 \sigma_x^2) \leq \frac{1}{k^2}. \quad (45)$$

اثبات: متغیر تصادفی $T = (X - \bar{X})^2$ را در نظر می گیریم. این متغیر فقط مقادیر مثبت را اختیار می کند. ضمناً می دانیم که $\bar{T} = \sigma_x^2$. از قسمت الف داریم:

$$P(T \geq \alpha) \leq \frac{\bar{T}}{\alpha}. \quad (46)$$

هرگاه به جای α در نامساوی اخیر قرار دهیم $k^2 \sigma_x^2$ بدست می آوریم:

$$P((X - \bar{X})^2 \geq k^2 \sigma_x^2) \leq \frac{\sigma_x^2}{k^2 \sigma_x^2} = \frac{1}{k^2}. \quad (47)$$

این نامساوی را به شکل زیر نیز می توان نوشت:

$$P(|X - \bar{X}| \geq k \sigma_x) \leq \frac{1}{k^2}, \quad (48)$$

پس از این لم به اثبات قضیه اولیه می پردازیم:

اثبات الف: احتمال اینکه یک رشته α متعارف نباشد را با P_0 نشان می دهیم. بنابراین تعریف داریم:

$$P_0 = Prob\{|\frac{f_i(\alpha) - mp_i}{\sqrt{mp_i(1-p_i)}}| \geq k, \text{ برای حداقل یک } i\} = \sum_{i=1}^N P(|\frac{f_i(\alpha) - mp_i}{\sqrt{mp_i(1-p_i)}}| \geq k). \quad (49)$$

با استفاده از نامساوی چبیشف نتیجه می گیریم که

$$P_0 \leq \sum_{i=1}^N \frac{1}{k^2} = \frac{N}{k^2} \leq \epsilon. \quad (50)$$

بنابراین قسمت الف قضیه ثابت می شود.

اثبات ب: فرض کنید که $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_m)$ یک رشته متعارف باشد. در این صورت با استفاده از تعریف داریم

$$mp_i - k\sqrt{mp_i(1-p_i)} \leq f_i(\alpha) \leq mp_i + k\sqrt{mp_i(1-p_i)}, \quad \forall i. \quad (51)$$

حال باتوجه به عدم همبستگی بین حروف، احتمال وقوع رشته α برابر است با:

$$P(\alpha) = p_1^{f_1(\alpha)} p_2^{f_2(\alpha)} \dots p_N^{f_N(\alpha)}. \quad (52)$$

از اینجا نتیجه می گیریم

$$\log P(\alpha) = \sum_{i=1}^N f_i(\alpha) \log p_i \quad (53)$$

و در نتیجه ترکیب با (51) بدست می آوریم:

$$\sum_{i=1}^N (mp_i - k\sqrt{mp_i(1-p_i)}) \log p_i \leq \log P(\alpha) \leq \sum_{i=1}^N (mp_i + k\sqrt{mp_i(1-p_i)}) \log p_i. \quad (54)$$

حال قرار می دهیم

$$A := -k \sum_{i=1}^N \sqrt{p_i(1-p_i)} \log p_i. \quad (55)$$

در نتیجه نامساوی قبلی به شکل زیر درمی آید:

$$-mH + A\sqrt{m} \leq \log P(\alpha) \leq -mH - A\sqrt{m}, \quad (56)$$

که از آن نتیجه می گیریم

$$2^{-mH-A\sqrt{m}} \leq P(\alpha) \leq 2^{-mH+A\sqrt{m}}. \quad (57)$$

اثبات پ: قبلاً نشان دادیم که احتمال اینکه یک رشته متعلق به زیرمجموعه رشته های متعارف باشد از $1 - \epsilon$ بیشتر است. از طرفی نشان دادیم که احتمال اینکه یک رشته خاص α ، رشته متعارف باشد از $2^{-mH-A\sqrt{m}}$ بیشتر است. پس اگر مجموعه رشته های متعارف را به صورت زیر در نظر بگیریم:

$$S := \{\alpha^{(1)}, \alpha^{(2)}, \alpha^{(3)}, \dots, \alpha^{(T)}\}, \quad (58)$$

که در آن T تعداد رشته های متعارف است، آنگاه داریم:

$$1 - \epsilon \leq P(\alpha^{(1)}) + P(\alpha^{(2)}) + \dots + P(\alpha^{(T)}) \leq T(2^{-mH+A\sqrt{m}}), \quad (59)$$

که از آن نتیجه می گیریم

$$2^{mH-A\sqrt{m}+\log(1-\epsilon)} \leq T, \quad (60)$$

و یا

$$2^{m(H+\delta_m)} \leq T, \quad (61)$$

که در آن $\lim_{m \rightarrow \infty} \delta_m = 0$. از طرفی می دانیم که

$$T2^{-mH-A\sqrt{m}} \leq P(\alpha^{(1)}) + P(\alpha^{(2)}) + \dots + P(\alpha^{(T)}) \leq 1, \quad (62)$$

که از آن نتیجه می گیریم

$$T \leq 2^{m(H+\frac{A}{\sqrt{m}})}. \quad (63)$$

از ترکیب (61، 63) نتیجه می گیریم که

$$T = 2^{m(H+\delta_m)} \quad (64)$$

که در آن $\lim_{m \rightarrow \infty} \delta_m = 0$. این همان نتیجه ای است که می خواستیم یعنی تعداد رشته های متعارف برابر است با 2^{mH} .

۸ نگاهی دوباره به اطلاعات شرطی و اطلاعات متقابل

بعد از فهم رشته های متعارف و فشرده سازی می توانیم نگاهی دوباره به اطلاعات شرطی و اطلاعات متقابل بیندازیم. از این زاویه جدید می توانیم تعریف متفاوتی برای تابع $H(X)$ پیدا کنیم. یاد گرفتیم که تعداد رشته های متعارف m حرفی برابر است با $2^{mH(X)}$. این حرف به این معناست که اگر کسی یک رشته معین را به عنوان سوال برای مادر نظر گرفته باشد و از ما بخواهد در یک مسابقه به اصطلاح « بیست سوالی » بپرسیدن سوال هایی که پاسخ آنها تنها آری یا خیر است به آن رشته معین دست پیدا کنیم در بهترین حالت می بایست تعداد $mH(X)$ بار سوال کنیم. زیرا بهترین نحوه سوال کردن نحوه ای است که در آن تعداد رشته های باقیمانده را به نصف مقدار قبلی کاهش می دهد و $2^{mH(X)}$ را به $2^{mH(X)-1}$ ، $2^{mH(X)-2}$ و سرانجام به ۱ تقلیل می دهد. به عنوان مثال منبع

$$X = \{000(1/2), 111(1/2)\} \quad (65)$$

که در آن اعداد داخل پرانتز احتمالات رشته ها را نشان می دهند. برای این منبع داریم $H(X) = 1$ می توانیم رشته سوالات خود را به ترتیب زیر تنظیم کنیم:

۱ - آیا همه اعداد صفر هستند؟

در هر دو صورت جواب آری یا خیر ما به رشته مورد نظری که سوال کننده در نظر گرفته است پی می بریم. یعنی یک سوال برای رسیدن به رشته مورد نظر کفایت می کند.

حال منبع زیر را در نظر بگیرید:

$$X = \{000(1/4), 111(1/4), 001(1/4), 110(1/4)\} \quad (66)$$

حال می توانیم سوالات خود را به شکل زیر تنظیم کنیم:

۱ - آیا اکثریت بیت ها صفر هستند؟

۲ - آیا همه بیت ها مثل هم هستند؟

در این صورت با دو سوال به رشته مورد نظری رسیدیم و $H(X)$ نیز برابر با ۲ است.

حال فرض کنید که فرستنده ای پیام های منبع X را از درون یک کانال ارسال می کند که در اثر نویز درون کانال این پیام ها به صورت Y توسط گیرنده دریافت می شوند. قبل از دریافت هر نوع پیامی اطلاعات منبع توسط آنتروپی $H(X)$ مشخص می شده است که نمایانگر حداقل سوالهای آری یا نه ای بوده است که بپرسیدن آنها می توانستیم به یک پیام خاص پی ببریم. وقتی که پیام های Y دریافت می شود آنتروپی منبع توسط تابع $H(X|Y)$ مشخص می شود که نشان دهنده تعداد جدید حداقل سوال

ه‌ا برای پی بردن به پیام اصلی است. هرگاه $H(X|Y) = 0$ باشد به معنای آن است به دقت می توانیم پیام منبع را تشخیص دهیم و در نتیجه اطلاعات متقابل Y و X زیاد است و هرگاه $H(X|Y) = H(X)$ باشد یعنی دانستن Y موجب هیچ کاهش در تعداد سوالات لازم نشده است. بنابراین تفاضل $H(X) - H(X|Y)$ یعنی تفاضل تعداد سوالات لازم برای شناسایی پیام های X ملاک خیلی خوبی از میزان اطلاعات متقابل X و Y است.

۹ ضمیمه: نحوه‌ی عملی ساختن کد کلمه ها برای یک منبع

در این ضمیمه که خواندن آن برای فهم درس های آینده ضروری نیست نحوه عملی ساختن کدهای یکتا گشا و فشرده را برای یک منبع توضیح می دهیم.

منبعی را در نظر می گیریم که پیام های یک آنزامل که توسط متغیر تصادفی X مشخص می شود را مخابره می کند. داریم

$$X = \{x_1, x_2, \dots, x_n\}. \quad (67)$$

در این آنزامل هر متغیر x_k با احتمال $P(x_k)$ اختیار می شود. از این به بعد x_k ها را کلمات می نامیم اگر چه ممکن است از تک حرف ها هجاها و یا جملات نیز تشکیل شده باشند. هر کلمه x_k را با یک کد متشکل از علائم 0 و 1 کدگذاری می کنیم. کدی که برای یک کلمه x به کار می بریم را با $w(x)$ نشان می دهیم و آن را یک کد کلمه یا (Code word) می خوانیم. رشته ای متشکل از کلمات را یک پیام می خوانیم و رشته ای متشکل از کد کلمه ها را یک کدپیام یا (Codemessage) می خوانیم. طول یک کد کلمه $w(x)$ را با $l(x)$ نشان می دهیم. هدف ما کمینه کردن طول متوسط کد کلمه ها یعنی عبارت زیر است:

$$\bar{l} := \sum_{i=1}^n P(x_i) l(x_i). \quad (68)$$

۱۰ یکتایی کد گشایی

اولین مسئله ای که با آن مواجه هستیم یکتایی کد گشایی است. برای مثال به جدول زیر توجه کنید: که در آن ستون سمت چپ کلمه ها و ستون سمت راست کد کلمه ها را نشان می دهد. حال هرگاه کد پیام 010 را دریافت کنیم می توانیم آن را به کدی برای هر کدام از پیام های x_2, x_3, x_1, x_4 تعبیر کنیم. در نتیجه این نوع کد گذاری دارای ابهام زیاد است و کد گذاری خوبی نیست. نخست باید یک صفت اساسی از هر نوع کدگذاری را مشخص کنیم.

x_1	0
x_2	010
x_3	01
x_4	10

مثالی از یک کد که در آن بعضی از کد کلمه های پیشوند کد کلمه های دیگرند 1:

x_1	0
x_2	100
x_3	101
x_4	11

مثالی از یک کد لحظه ای 2:

تعریف: یک کد یکتا گشتا اگر هر کد پیام حداکثر متناظر با یک پیام باشد.

یک راه برای نوشتن کد های یکتا گشتا آن است که تقاضا کنیم هیچ کد کلمه ای پیشوند کد کلمه دیگری نباشد.

تعریف: یک کد کلمه A پیشوند یک کد کلمه B خوانده می شود اگر B را بتوان به صورت $B = AC$ نوشت که در آن C دلخواه است و لزومی ندارد که خود یک کد کلمه باشد. در جدول (۱۰) x_1 پیشوند x_2 و x_3 است. x_3 نیز پیشوند x_2 است.

تعریف: یک کد که در آن هیچ کد کلمه ای پیشوند کد کلمه دیگری نباشد یک کد لحظه ای خوانده می شود.

مثال: کد زیر یک کد لحظه ای است.

نکته مهم در مورد این نوع کد ها آن است که هر کد لحظه ای یکتا گشتا است. البته معکوس این قضیه درست نیست. باز هم به کد نشان داده شده در جدول ?? دقت کنید. هرگاه کد پیامی مثل رشته

$$101110100101 \quad (69)$$

را دریافت کنیم تنها می توانیم آن را به صورت پیام زیر بازگشایی کنیم:

$$x_3 \ x_4 \ x_1 \ x_2 \ x_3. \quad (70)$$

حال کد زیر را در نظر بگیرید:

این کد لحظه ای نیست زیرا x_1 پیشوند x_2 است. با این وجود این کد به طور یکتا گشوده می شود. زیرا هر رشته ای را که دریافت می کنیم رشته ای از 0 ها است که در بعضی جاهای آن 1 های منفرد قرار گرفته اند، مثل رشته زیر:

$$001000101010000001. \quad (71)$$

x_1	0
x_2	01

یک کد که به طور یکتا گشوده می شود ولی لحظه ای نیست. 3:

x_1	0
x_2	010
x_3	01
x_4	10

مثالی از یک کد که به طور یکتا گشوده نمی شود. 4:

چنین رشته ای به آسانی قابل گشایش است و کدی برای پیام زیر است:

$$x_1 x_2 x_1 x_1 x_2 x_2 x_2 x_1 x_1 x_1 x_1 x_2. \quad (72)$$

در زیر روشی را بیان می کنیم که به کمک آن می توانیم تشخیص بدهیم که آیا یک کد به صورت یکتا گشوده می شود یا خیر.

فرض کنید که S_0 مجموعه همه کد کلمه ها باشد. مجموعه تمام پسوندهایی را که در S_0 وجود دارد در مجموعه دیگری به نام S_1 قرار می دهیم. حال مجموعه $S_2, S_3, \dots, S_n$ را به طریق زیر تشکیل می دهیم:

الف: اگر یک کد کلمه $A \in S_0$ پیشوند کد کلمه ای مثل $w = AB \in S_{n-1}$ باشد، B را در S_n قرار می دهیم.

ب: اگر یک کد کلمه $A \in S_{n-1}$ پیشوند کد کلمه ای مثل $w = AB \in S_0$ باشد، B را در S_n قرار می دهیم.

قضیه: یک کد به صورت یکتا گشوده می شود اگر و فقط اگر $S_0 \cap [S_1 \cup S_2 \cup S_3 \dots] = \emptyset$

مثال: کد زیر یکتا گشایش نیست.

زیرا:

$$S_0 = \{0, 010, 01, 10\} \quad S_1 = \{10, 1, 0\} \quad (73)$$

$$S_0 \cap S_1 \neq \emptyset$$

مثال: کد زیر یکتا گشایش است:

x_1	0
x_2	001

مثالی از یک کد یکتاگشا 5:

زیرا:

$$S_0 = \{0, 001\} \quad S_1 = \{01\} \quad S_2 = \{1\}, \quad (74)$$

$$S_0 \cap [S_1 \cup S_2] = \phi \text{ و}$$

مثال: کد زیر را درنظرمی گیریم:

x_1	a
x_2	c
x_3	ad
x_4	abb
x_5	bad
x_6	deb
x_7	bbcde

(75)

برای این کد داریم:

$$\begin{aligned} S_0 &= \{a, c, ad, abb, bad, deb, bbcde\} \\ S_1 &= \{d, bb\} \\ S_2 &= \{eb, cde\} \\ S_3 &= \{de\} \\ S_4 &= \{b\} \\ S_5 &= \{ad, bcde\} \\ S_6 &= \{d\} \\ S_7 &= \{eb\} \end{aligned} \quad (76)$$

با توجه به این روابط خواهیم دید که

$$S_0 \cap [S_1 \cup S_2 \cup S_3 \cup S_4 \cup S_5 \cup S_6 \cup S_7] = \phi, \quad (77)$$

و درنتیجه این کد یکتاگشا است.

قضیه (شرط لازم و کافی برای وجود کد های لحظه ای):

مجموعه کلمه های $X = \{x_1, x_2, \dots, x_M\}$ و مجموعه حروف الفبای $A := \{a_1, a_2, \dots, a_D\}$ داده شده اند. مجموعه اعداد صحیح $\{n_1, n_2, \dots, n_M\}$ نیز مفروض اند. آیا یک کد لحظه ای می توان از الفبای A نوشت که طول های $\{n_1, n_2, \dots, n_M\}$ داشته باشند؟ پاسخ این سوال مثبت است اگر فقط اگر شرط زیر برقرار باشد:

$$\sum_{i=1}^M \frac{1}{D^{n_i}} \leq 1. \quad (78)$$

این نامساوی به نامساوی *Kraft* مشهور است.

قبل از اثبات این قضیه به یک نتیجه ساده آن توجه می کنیم:

نتیجه: برای حروف الفبای انگلیسی با احتساب نقطه، کاما، و دیگر علائم داریم: $M = 32$. هم چنین اگر بخواهیم از الفبای $\{0, 1\}$ استفاده کنیم داریم $D = 2$. بنابراین باید داشته باشیم:

$$\sum_{i=1}^{32} \frac{1}{2^{n_i}} \leq 1 \longrightarrow n_{min} \geq 5. \quad (79)$$

بنابراین نمی توان هیچ حرفی را با کمترین 5 بیت کد کرد.

حال به اثبات قضیه می پردازیم:

اثبات: می توانیم از نمودارهای درختی استفاده کنیم. یک درخت با مرتبه D و اندازه k درختی است که D ریشه دارد و از هر ریشه نیز D شاخه منشعب می شود و این کار ادامه می یابد تا $k-1$ مرحله. در این صورت تعداد شاخه های آخرین مرحله عبارت است از D^k . ریشه اول درخت متناسب با کد کلمه های تک حرفی $\{1, 2, 3, \dots, D\}$ هستند. شاخه های مرحله بعد متناسب با کد کلمات دو حرفی هستند مثل $\{11, 12, \dots, DD\}$ و همینطور تا آخر. به این ترتیب هر کد کلمه متناسب با یکی از گره های این درخت می شود. حال اگر بخواهیم یک کد لحظه ای بسازیم می بایست کد کلمه های خود را از شاخه های این درخت به نحو خاصی انتخاب کنیم. هر کد کلمه یا هر گره که از این درخت انتخاب می کنیم می بایست تمام شاخه های منشعب از آن گره را کنار بگذاریم زیرا همه کد کلمه های مربوط به آن شاخه ها کلمه مربوط به این گره را به عنوان پیشوند خود دارند. اگر طول یک کد کلمه که انتخاب می کنیم برابر با i باشد، تعداد شاخه هایی که از آن منشعب می شود برابر است با D^{k-i} . بنابراین به ازای هر کد کلمه به طول i تعداد D^{k-i} تا از شاخه ها حذف می شوند. در نتیجه خواهیم داشت:

$$D^{k-i_1} + D^{k-i_2} + \dots + D^{k-i_M} \leq D^k \quad (80)$$

که با تقسیم طرفین بر D^k به رابطه (90) می رسیم. این رابطه را می توان به شکل زیر نیز نوشت:

$$\sum_i w_i D^{-i} \leq 1, \quad (81)$$

که در آن w_i تعداد کد کلمه های با طول i است.

حال معکوس قضیه را ثابت می کنیم: تاکنون ثابت کردیم که اگر کد لحظه ای باشد می بایست شرط (90) برقرار باشد. حال نشان می دهیم که به ازای هر $(n_1, n_2, \dots, n_M)$ که در شرط (90) صدق کند می توان یک کد لحظه ای ساخت. n_i را به شکل زیر مرتب می کنیم:

$$n_1 \leq n_2 \leq n_3 \leq \dots \leq n_M. \quad (82)$$

حال یک نقطه به اندازه n_1 را روی درخت با مرتبه D و اندازه n_M انتخاب می کنیم. به این ترتیب تعداد $D^{n_M-n_1}$ نقطه حذف می شوند. تعداد نقاط باقیمانده برابر است با $D^{n_M} - D^{n_M-n_1}$. نقطه دوم را به طول n_2 انتخاب می کنیم. این نقطه تعداد $D^{n_M-n_2}$ نقطه دیگر را حذف می کند. تعداد نقاط باقیمانده برابر است با $D^{n_M} - D^{n_M-n_1} - D^{n_M-n_2}$. این کار را ادامه می دهیم تا نقطه ماقبل آخر که طول آن n_{M-1} است. این نقطه نیز تعداد $D^{n_M-n_{M-1}}$ را حذف می کند. آیا درخت موردنظر این همه جا دارد؟ برای پاسخ به این سوال کافی است که تعداد نقاط باقیمانده را بعد از مرحله ماقبل آخر بشماریم: این تعداد برابر است با

$$\begin{aligned} Q &= D^{n_M} - D^{n_M-n_1} - D^{n_M-n_2} - \dots - D^{n_M-n_{M-1}} \\ &= D^{n_M} [1 - D^{-n_1} - D^{-n_2} - \dots - D^{-n_{M-1}}] \end{aligned} \quad (83)$$

اما چون شرط (90) برقرار است خواهیم داشت:

$$\sum_{i=1}^M D^{-n_i} \leq 1 \longrightarrow 1 - \sum_{i=1}^{M-1} D^{-n_i} \leq D^{-n_M} \quad (84)$$

و این رابطه به این معناست که حداقل یک انتخاب برای آخرین کد کلمه باقی می ماند. اثبات قضیه در این جا کامل می شود.

قضیه: نامساوی کرافت شرط لازم و کافی برای ساختن کد های یکتاگشاست.

اثبات الف: اگر نامساوی کرافت برقرار باشد می توانیم یک کد لحظه ای مطابق با قضیه قبل بسازیم و می دانیم که کد های لحظه ای یکتا گشاهستند.

اثبات ب: حال فرض کنید که یک کد یکتا گشاداریم. می خواهیم نشان دهیم که حتماً نامساوی کرافت برقرار است. بجای عبارت $\sum_i D^{-n_i}$ عبارت $\sum_{i=1}^r w_i D^{-i}$ را بکار می بریم که در آن w_i تعداد کلمات با طول i است. حال عبارت اخیر را می توان به صورت یک تابع مولد تعبیر کرد. می توان دریافت که

$$\left(\sum_{i=1}^r w_i D^{-i} \right)^n = \sum_{k=r}^{nr} X_k D^{-k}, \quad (85)$$

که در آن X_k تعداد کل کد کلمه های با طول k در کدگذاری رشته های r تایی است. اما می دانیم که کد از نوع یکتا گشودنی است. در ضمن تعداد کل کد کلمه های با طول k برابر است با D^k . چون کد های یکتا گشا زیر مجموعه کلیه کد ها هستند نتیجه می گیریم که $X_k \leq D^k$. بنابراین خواهیم داشت:

$$\left(\sum_{i=1}^r w_i D^{-i} \right)^n \leq \sum_{k=r}^{nr} 1 = nr - r + 1. \quad (86)$$

و از آنجا

$$\left(\sum_{i=1}^r w_i D^{-i} \right) \leq (1 + (n-1)r)^{\frac{1}{n}}. \quad (87)$$

در حد n های بزرگ این رابطه تبدیل می شود به نامساوی کرافت.

قضیه کدگذاری بدون نوفه: مجموعه کلمات $X = \{x_1, x_2, \dots, x_M\}$ که در آن نماد x_i با احتمال $P_i := P(x_i)$ ظاهر می شود و مجموعه حروف الفبای $A := \{a_1, a_2, \dots, a_D\}$ داده شده اند. این نماد ها با کد کلمه های $\{w_1, w_2, \dots, w_M\}$ کد شده اند و طول هر کد کلمه w_i برابر است با $n_i := l(w_i)$. هدف ما آن است که طول متوسط کد کلمه ها را کمینه کنیم یعنی کمیت زیر را:

$$\bar{n} := \sum_{i=1}^M p_i n_i. \quad (88)$$

مجموعه اعداد صحیح $\{n_1, n_2, \dots, n_M\}$ نیز مفروض اند. بهترین کد یکتا گشایی که می توان برای کد کردن این الفبا ساخت، یعنی کد یکتا گشایی که کمترین طول متوسط را داشته باشد کدی است با طول متوسط

$$\bar{n} = \frac{H(X)}{\log D}. \quad (89)$$

اثبات: نخست توجه می کنیم که کد مورد نظر مایکتا گشاست اگر و فقط اگر شرط زیر برقرار باشد:

$$\sum_{i=1}^M \frac{1}{D^{n_i}} \leq 1. \quad (90)$$

بقیه اثبات را در سه مرحله انجام می دهیم. از این به بعد نیز ما فقط درباره کد های یکتا گشا حرف می زنیم. در مرحله اول یک حد پایین برای $\bar{n}$ پیدا می کنیم و نشان می دهیم که

$$\bar{n} \geq \frac{H(X)}{\log D} \quad (91)$$

که در آن شرط تساوی برقرار می شود اگر فقط اگر $p_i = D^{-n_i}$.
 در مرحله دوم تحقیق می کنیم که چقدر می توانیم به این حد پایین نزدیک شد. و بالاخره در مرحله سوم بهترین کد ممکن را می سازیم.
 برای اثبات نامساوی (91) می بایست نامساوی زیر را ثابت کنیم:

$$\sum_{i=1}^M n_i p_i \geq - \sum_{i=1}^M p_i \frac{\log p_i}{\log D}, \quad (92)$$

و یا

$$\sum_{i=1}^M (n_i \log D) p_i \geq - \sum_{i=1}^M p_i \log p_i. \quad (93)$$

قبلا داشتیم که به ازای هر دو توزیع احتمال $\{p_i\}$ و $\{q_i\}$ ، نامساوی زیر برقرار است:

$$\sum_i -p_i \log p_i \leq - \sum_i p_i \log q_i, \quad (94)$$

و تساوی تنها وقتی برقرار می شود که $\{q_i\} = \{p_i\}$.

می توانیم یک توزیع احتمال مطابق با رابطه زیر تعریف کنیم:

$$q_i := \frac{D^{-n_i}}{\sum_{i=1}^M D^{-n_i}} \quad (95)$$

و از رابطه (92) استفاده کنیم. یک محاسبه ساده منجر به رابطه زیر خواهد شد:

$$H(X) \leq \bar{n} \log D + \log \left(\sum_{i=1}^M D^{-n_i} \right), \quad (96)$$

که تساوی وقتی برقرار می شود که

$$p_i = \frac{D^{-n_i}}{\sum_{i=1}^M D^{-n_i}}. \quad (97)$$

حال باتوجه به اینکه برای کدهای یکتاگشانا نامساوی کرافت برقرار است یعنی $\sum_{i=1}^M D^{-n_i} \leq 1$ نتیجه می گیریم که $\log \sum_{i=1}^M D^{-n_i} \leq 0$ و از آنجا بدست می آوریم که

$$H(X) \leq \bar{n} \log D. \quad (98)$$

هرگاه بتوانیم یک کد را چنان انتخاب کنیم که طول کد کلمه های آن از رابطه $n_i = \log_D \frac{1}{p_i}$ تبعیت کند، آنگاه خواهیم داشت :
 $\bar{n} = \frac{H(X)}{\log D}$. معکوس این قضیه نیز صحیح است یعنی اینکه اگر رابطه $\bar{n} = \frac{H(X)}{\log D}$ برقرار باشد آنگاه $p_i = D^{-n_i}$ برای اثبات
 این نتیجه از رابطه 96 استفاده می کنیم و به این نتیجه می رسیم که

$$\bar{n} \log D \leq \bar{n} \log D + \log \left(\sum_{i=1}^M D^{-n_i} \right), \quad (99)$$

و از آنجا باتوجه به اینکه $\sum_{i=1}^M D^{-n_i} \leq 1$ ، به این نتیجه می رسیم که $\sum_{i=1}^M D^{-j} = 1$. اما باتوجه به رابطه 97 این نتیجه به
 این معناست که $p_i = D^{-n_i}$

تعریف: یک کد کاملاً بهینه کدی است که برای آن $\bar{n} = \frac{H(X)}{\log D}$
 یک مثال از یک کد کاملاً بهینه در جدول زیر داده شده است:

X	P	Cw
x_1	$\frac{1}{2}$	0
x_2	$\frac{1}{4}$	10
x_3	$\frac{1}{8}$	110
x_4	$\frac{1}{8}$	111

(100)

این کد دارای این خاصیت است که $n_i = \log \frac{1}{p_i}$
 در حالت کلی معلوم نیست که بتوان کد را چنان طراحی کرد که حد $\bar{n} = \frac{H}{\log D}$ برقرار شود، زیرا اعداد $n_i = \log_D \frac{1}{p_i}$ معلوم نیست
 که صحیح باشند. با این وجود می توان کاری کرد که شرط زیر برقرار شود:

$$\log_D \frac{1}{p_i} \leq n_i \leq \log_D \frac{1}{p_i} + 1. \quad (101)$$

در این صورت خواهیم داشت :

$$\frac{H(X)}{\log D} \leq \bar{n} \leq \frac{H(X)}{\log D} + 1. \quad (102)$$

حال نکته این است که هر قدر بخواهیم می توانیم به حدّ پایین نامساوی بالا نزدیک شویم. برای این کار می بایست از کدهای
 چندتایی یا کدهای بلوکی استفاده کنیم. فرض کنید به جای کد نگاری X رشته های s تایی از X ها را کد نگاری کنیم، یعنی
 رشته های $Y = (X_1, X_2, \dots, X_s)$ را. حال باید نشان دهیم که تحت این شرایط طول کد کلمه ها به ازای هر X پایین می آید.

به رابطه (101) دقت می کنیم. از آنجا که $Y = (X_1, X_2, \dots, X_s)$ ، کلمه ها به صورت s تایی های از نوع $y_{ij} = (x_i, x_j, \dots, x_s)$ هستند. داریم

$$H(Y) = - \sum_{i,j,\dots} p_{ij,\dots} \log p_{ij,\dots} \quad (103)$$

چون کلمات پیام Y از هم مستقل هستند خواهیم داشت: $H(Y) = sH(X)$. و بنابراین

$$\frac{H(Y)}{\log D} \leq \bar{n} \leq \frac{H(Y)}{\log D} + 1, \quad (104)$$

و یا

$$\frac{H(X)}{\log D} \leq \frac{1}{s} \bar{n} \leq \frac{H(X)}{\log D} + \frac{1}{s}. \quad (105)$$

در این رابطه $\frac{1}{s} \bar{n}$ طول متوسط هر کد کلمه به ازای هر کلمه در X است و در حد s های بزرگ دیده می شود که ما به حد بهینه نزدیک می شویم.

۱۱ ساختن کد های بهینه

حال باید الگوریتمی را معرفی کنیم که کد های بهینه را به طور روشمند می سازد. نخست به یک لم احتیاج داریم:

لم: فرض کنید که برای احتمالات $P_1, P_2, \dots, P_M$ ، یک کد C در درون مجموعه کد های لحظه ای بهینه باشد. یعنی هیچ کد لحظه ای دیگری با طول متوسط کمتر از طول متوسط مربوط به C وجود نداشته باشد. در این صورت این کد در درون مجموعه کد های یکتا گشا نیز بهینه است.

اثبات: می دانیم که کد های لحظه ای زیرمجموعه کد های یکتا گشا است. حال فرض کنید که یک کد یکتا گشای C' با طول کد کلمه های $n'_1, n'_2, \dots, n'_M$ وجود دارد که طول متوسط آن از طول متوسط C کمتر است. اولاً چون C' یکتا گشا است بنابراین قضیه ای که قبلاً ثابت کردیم خواهیم داشت: $\sum_{i=1}^M D^{-n'_i} \leq 1$. اما در این صورت بنابراین قضیه قبل یک کد لحظه ای با طول کلمات $n'_1, n'_2, \dots, n'_M$ وجود خواهد داشت. بدین ترتیب بهینه بودن کد C در درون مجموعه کد های لحظه ای نیز نقض می شود.

از این به بعد توجه خود را به کد های لحظه ای و دوتایی *binary* معطوف می کنیم. نخست به یک لم احتیاج داریم:

لم: فرض کنید که C یک کد لحظه ای با طول کد کلمه های $n_1, n_2, \dots, n_M$ برای کد گذاری علامات $x_1, x_2, \dots, x_M$ باشد که این علامات نیز با احتمالات $p_1, p_2, \dots, p_M$ تکرار شوند. در این صورت اگر کد C درون کد های لحظه ای بهینه باشد آنگاه خاصیت های زیر برقرارند:

الف: علامت های با احتمال بیشتر طول کمتر دارند. یعنی اگر $p_i \geq p_j$ آنگاه $n_i \leq n_j$.

ب: دوتا از کد کلمه هایی که کمترین احتمال ها را دارند حتماً دارای طول مساوی هستند.

پ: در بین کلماتی که بیشترین طول را دارند، حتماً باید دو کلمه وجود داشته باشند که فقط و فقط در یک رقم بایکدیگر تفاوت داشته باشند.

اثبات الف: فرض کنید که $p_1 \geq p_2$ که در آن p_2, p_1 به ترتیب احتمال ظهور علامات x_2, x_1 باشند. هم چنین فرض کنید که در این کد لحظه ای C داشته باشیم $n_1 \geq n_2$. در این صورت می توان یک کد بهتر از C ساخت. جای کد کلمه های مربوط به x_1 و x_2 را عوض می کنیم. کد هنوز لحظه ای است زیرا شرط کرافت برقرار است. در کد جدید C' داریم:

$$\bar{n}' - \bar{n} = n_1 p_2 + n_2 p_1 - n_1 p_1 - n_2 p_2 = (n_1 - n_2)(p_2 - p_1) \leq 0. \quad (106)$$

اثبات ب: فرض کنید که کمترین احتمالات عبارت باشند از P_{M-1}, P_M و $P_{M-1} \geq P_M$. حال می خواهیم حالت $n_{M-1} < n_M$ را حذف کنیم. کد کلمه های مربوط به علامت های x_M و x_{M-1} را به ترتیب با S و $\tilde{S}$ نشان می دهیم. فرض کنید که

$$\begin{aligned} S &= s_1 s_2 \dots s_{n_{M-1}} \\ \tilde{S} &\equiv S' \tilde{S}' = s'_1 s'_2 \dots s'_{n_{M-1}} (s'_{n_{M-1}+1} s_{n_{M-1}+2} \dots s'_{n_M}) \end{aligned} \quad (107)$$

حال می توانیم قسمت اضافی را که در پیرانتز قرار داده ایم حذف کنیم بدون اینکه به لحظه ای بودن کد خللی وارد شود. چون اگر کلمه ای پیشوند $S' \tilde{S}'$ نبوده است پیشوند S' نیز نخواهد بود. ضمناً S' نمی تواند پیشوند کد کلمه دیگری باشد، چون کلمات مربوط به x_M و x_{M-1} بزرگترین طول ها را دارند. تنها مکانی که باقی می ماند آن است که کلمات با طول n_{M-1} بیش از دو تاباشند. در این صورت تنها راه برای پیشوند بودن S' آن است که S' دقیقاً بایکی از آن کلمات برابر باشد. ولی این بدان معناست که در کد اولیه که در آن حذفی صورت نگرفته بود، آن کلمه خاص پیشوند $\tilde{S}$ بوده است.

اثبات پ: حال فرض کنید که دوتا از بلندترین کلمات را در نظر بگیریم. اگر تنها در رقم آخر اختلاف داشته باشند که این همان چیزی است که مطلوب ماست. اگر بیش از رقم آخر با هم اختلاف داشته باشند ما می توانیم رقم آخر را حذف کنیم و یک کد لحظه ای بهتر بدست بیاوریم. استدلال این که لحظه ای بودن کد به هم نمی خورد مثل قسمت ب است.

۱۲ روش هوفمان برای ساختن کد های لحظه ای بهینه

ازاین به بعد نمادها واحتمالات را با (X, P) نمایش می دهیم:

$$(X, P) = \{(x_1, p_1), (x_2, p_2), \dots (x_M, p_M)\}. \quad (108)$$

مرحله اول: از (X, P) یک $(\tilde{X}, \tilde{P})$ به ترتیب زیرمی سازیم:

$$(\tilde{X}, \tilde{P}) = \{(x_1, p_1), (x_2, p_2), \dots (x_{M-2}, p_{M-2}), (x_{M-1,M}, p_{M-1} + p_M)\}. \quad (109)$$

سوال: منظور از $x_{M-1,M}$ چیست؟ منظوراین است که درذهن خود تفاوت بین x_M و x_{M-1} را ازبین ببریم. به عبارت دیگر می دانیم که تنها احتمالات مهم هستند و نه خود نمادها. بنابراین مجموعه $\{p_1, p_2, \dots p_{M-1}, p_M\}$ را به مجموعه $\{p_1, p_2, \dots p_{M-1} + p_M\}$ تقلیل داده ایم. حال فرض کنید که کد بهینه ای برای $(\tilde{X}, \tilde{P})$ در دست باشد با مشخصات زیر:

$\tilde{X}$	$\tilde{P}$	$\tilde{C}$	$\tilde{N}$
x_1	p_1	w_1	n_1
x_2	p_2	w_2	n_2
.	.	.	.
x_{M-2}	p_{M-2}	w_{M-2}	n_{M-2}
$x_{M-1,M}$	$p_{M-1} + p_M$	$w_{M-1,M}$	$n_{M-1,M}$

(110)

حال کد C را برای (X, P) به شکل زیرمی سازیم.

$\tilde{X}$	$\tilde{P}$	$\tilde{C}$	$\tilde{N}$
x_1	p_1	w_1	n_1
x_2	p_2	w_2	n_2
$\cdot$	$\cdot$	$\cdot$	$\cdot$
x_{M-2}	p_{M-2}	w_{M-2}	n_{M-2}
x_{M-1}	p_{M-1}	$w_{M-1,M}0$	$n_{M-1,M} + 1$
x_M	p_M	$w_{M-1,M}1$	$n_{M-1,M} + 1$

(111)

حال ثابت می کنیم که اگر $\tilde{C}$ بهینه باشد آنگاه C نیز بهینه است. ازبرهان خلف استفاده می کنیم. فرض کنید که کدی مثل C' وجود داشته باشد که از کد C بهتر باشد. دراین صورت بااستفاده از کد C' می توان کدی مثل $\tilde{C}'$ ساخت که از $\tilde{C}$ بهتر باشد. کد C' درجدول زیرنشان داده شده است:

X	P	C'	N'
x_1	p_1	w'_1	n'_1
x_2	p_2	w'_2	n'_2
$\cdot$	$\cdot$	$\cdot$	$\cdot$
x_{M-2}	p_{M-2}	w'_{M-2}	n'_{M-2}
x_{M-1}	p_{M-1}	w'_{M-1}	n'_{M-1}
x_M	p_M	w'_M	n'_M

(112)

دراین کد $n'_m = n'_{M-1}$. w'_{M-1} و w'_M نیزتنها دررقم آخرباهم اختلاف دارند. حال کد $\tilde{C}'$ را مطابق جدول زیرمی سازیم:

X	P	$\tilde{C}'$	$\tilde{N}'$
x_1	p_1	w'_1	n'_1
x_2	p_2	w'_2	n'_2
$\cdot$	$\cdot$	$\cdot$	$\cdot$
x_{M-2}	p_{M-2}	w'_{M-2}	n'_{M-2}
$x_{M-1,M}$	$p_{M-1} + p_M$	$\tilde{w}'_{M-1,M}$	n'_{M-1}

(113)

که در آن $\tilde{w}'_{M-1,M}$ همان w'_M یا w'_{M-1} است که رقم آخر آن برداشته شده است. حال بدست می آوریم:

$$\bar{n} - \bar{\tilde{n}} = (p_{M_1} + p_M)(n_{M-1,M} + 1 - n_{M-1,M}) = p_{M-1} + p_M, \quad (114)$$

و

$$\overline{n'} - \overline{\tilde{n}'} = (p_{M_1} + p_M)(n'_M - n'_M - 1) = p_{M-1} + p_M. \quad (115)$$

در نتیجه

$$\bar{n} - \bar{\tilde{n}} = \overline{n'} - \overline{\tilde{n}'} \quad (116)$$

که از آن خواهیم داشت:

$$if \quad \overline{n'} < \bar{n} \longrightarrow \overline{\tilde{n}'} < \bar{\tilde{n}}. \quad (117)$$

بنابراین اگر کد C' از کد C بهتر باشد کد $\tilde{C}$ نیز از کد $\tilde{C}$ بهتر است و این خلاف بهینه بودن کد $\tilde{C}$ است.

این قضایا به ما می آموزند که چگونه کد های بهینه بسازیم.
مثال یک: روش ساخت در جدول های زیر نشان داده شده است:

X	P
x_1	0.5
x_2	0.35
x_3	0.15

(118)

$\tilde{X}$	$\tilde{P}$	$\tilde{C}$
x_1	0.5	0
$x_{2,3}$	0.5	1

(119)

واز آنجا

X	P	C
x_1	0.5	0
x_2	0.35	10
x_3	0.15	11

(120)

هرگاه تعداد کلمات بیشتر باشد این کار را در چند مرحله انجام می دهیم . در هر مرحله احتمالات را از بیشترین به کمترین مرتب می کنیم و آخرین دو کلمه را با هم مطابق با آنچه که در بالا گفته شد ادغام می کنیم. این کار را آنقدر انجام می دهیم تا به یک مجموعه برسیم متشکل از دو نماد و دو احتمال. به دو نماد آخر کلمه های 0 و 1 را نسبت می دهیم و سپس مراحل را در جهت عکس طی می کنیم تا به جدول اولیه برسیم و کد های تمام نماد ها را بدست آوریم.