Simonyi Gábor
Információközlés és gráfelmélet

A 2009. szeptember 29-i előadás kibővített változata
Lejegyezte és szerkesztette Lovas Lia Izabella

Bevezető feladat: Valaki kiválasztja egy sakktábla egy tetszőleges mezőjét. Legalább hány eldöntendő kérdésre van szükségünk ahhoz, hogy biztosan kitalálhassuk a gondolt mezőt?

A feladat megoldása igen egyszerű: 6 kérdés elég, ugyanis minden lépésben feloszthatjuk két egyenlő részre a még szóba jöhető mezőket, és rákérdezhetünk, hogy a keresett mező melyik csoportban található. Ilyen módon a hatodik kérdés után egyetlen mező marad. Másrészt hatnál kevesebb kérdés nem lehet elég: ha a még ki nem zárt mezőket következő kérdésünk két nem egyenlő csoportra bontja, akkor mindig lehetséges, hogy a kiválasztott mező a nagyobb elemszámú halmazba kerül.

Felvetődik: vajon akkor is 6-e a fenti feladat megoldása, ha előre meg kell adnunk az összes kérdésünket, és csak ezután kapjuk meg a válaszokat?

Megoldás: Igen. Képzeljük el, hogy minden mezőhöz hozzárendelünk egy 6 karakterből álló 0-1 sorozatot. Ezekből éppen $2^6 = 64$ különböző létezik, így a sakktábla minden mezejéhez különböző sorozatot rendelhetünk. Első kérdésünk az lehet, 1-es-e a mezőhöz tartozó sorozat első eleme, majd ugyanígy végigmehetünk a sorozat összes bitjén. A hatodik kérdés után ismerni fogjuk a mezőhöz rendelt teljes sorozatot, tehát magát a mezőt is kitaláltuk.

A fenti egyszerű példában a sakktábla mezőihez 0-1 sorozatokat rendeltünk, lényegében kódoltuk őket. Egy-egy ilyen 0-kból és 1-esekből álló sorozatot a későbbiekben *bináris kódszó*nak fogunk nevezni.

Az információelmélet születése Claude Shannon nevéhez fűződik, aki 1948-ban megjelent A Mathematical Theory of Communication című munkájában lefektette a matematika ezen új területének alapjait. A későbbi évek jelentős eredményei közül is számos az ő nevéhez fűződik. Ezen írás keretein belül csak arra van lehetőségünk, hogy rövid ízelítőt adjunk az információelmélet alapfogalmaiból, illetve vázlatosan rávilágítsunk egy érdekes kapcsolatra a gráfelmélettel.

Ajánló
· A Mathematical Theory of Communication:
http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html
· Aaron D. Wyner: Shannon művének jelentősége:
http://cm.bell-labs.com/cm/ms/what/shannonday/work.html
· Claude Shannon a MacTutor Matematikatörténeti Gyűjteményben:
http://www-groups.dcs.st-and.ac.uk/~history/Biographies/Shannon.html
· Claude Shannon – Father of the information Age (film):
http://www.youtube.com/watch?v=z2Whj_nL-x8

Egy adott üzenet információtartalmát a kódolásához minimálisan szükséges bitek számával fogjuk mérni. Érdemes megfigyelni, hogy ez teljesen független az üzenet tartalmától, amitől függ, az a lehetséges üzenetek száma. Adott kommunikációs helyzetben törekszünk arra, hogy minél rövidebb üzenetet küldjünk. Pontosabban fogalmazva, azt akarjuk elérni, hogy üzenetünk várható hossza minimális legyen. Figyelembe szeretnénk venni, hogy egy esemény lehetséges kimenetelei közül nem biztos, hogy mindegyik azonos valószínűséggel következik be. Ilyenkor nem biztos, hogy minden kimenetelt érdemes azonos hosszúságú kódszavakkal kódolni. Vegyünk egy példát:

Minden héten lottózunk, és hónap végén (azaz négyhetenként; az egyszerűség kedvéért feltesszük, hogy minden hónap 4 hétből áll) szeretnénk egy üzenetben elküldeni, melyik héten nyertünk, és melyiken nem. Ez nyilván megoldható, ha 4 hosszúságú bináris kódot alkalmazunk: azokhoz a hetekhez, amikor nem nyertünk, 0-t rendelünk, ellenkező esetben 1-et. Ez a módszer nem túl gazdaságos: ha az eljárást minden hónap végén megismételjük, az esetek döntő többségében a 0000 sorozatot fogjuk elküldeni. Ehelyett küldhetünk pl. egyetlen 0 bitet, a többi, nagyon kis valószínűségű esetet pedig 1 bitnél hosszabb sorozatokkal kódoljuk. Ha az üzenetek küldését hosszú időn át folytatjuk, átlagosan nyilván 4-nél jóval kevesebb bitet kell elküldenünk havonta.

A továbbiakban is bináris kódokkal foglalkozunk. Vizsgáljuk azt az esetet, amikor n lehetséges kimenetel van, az i-edik kimenetel valószínűsége p_i, a hozzá rendelt kódszó hossza pedig l_i. Célunk, hogy a $\sum_{i=1}^{n} p_i \cdot l_i$ összeget, ami a küldött üzenet átlagos hossza (másként fogalmazva: az üzenet hosszának várható értéke), minimalizáljuk.

Meg fogjuk mutatni, hogy $\sum_{i=1}^{n} p_i \cdot l_i$ alulról becsülhető a $P=(p_1,p_2,p_3,\dots,p_n)$ valószínűségeloszlás entrópiájával, melyet az alábbi módon definiálunk (p_i=0 esetén a megfelelő tagot 0-nak vesszük):

Entrópia:

$$H(P)=\sum_{i=1}^{n} p_i \cdot \log_2 \frac{1}{p_i}$$

Szokás ezt úgy értelmezni, hogy a p_i valószínűségű esemény bekövetkezésének információtartalma $\log_2 \frac{1}{p_i}$, és így az adott valószínűségi változó értékei átlagosan $H(P)$ információt hordoznak. Az alábbi egyszerű tények azt mutatják, hogy ez az értelmezés összhangban van néhány természetes elvárással:

- Biztos esemény bekövetkezése nem ad információt, így elvárjuk, hogy a p_i=1-hez tartozó esemény információtartalma 0 legyen. $\log_2 1 = 0$ valóban teljesül.
- Két egyforma valószínűségű esemény egyikének bekövetkezte jelentsen 1 bit információt. Ennek a feltételnek is megfelel a fenti értelmezés: $\log_2 \frac{1}{0{,}5} = 1$.
- Egymástól független események együttes bekövetkezésének információtartalma egyezzen meg az egyes események bekövetkezte által hordozott információtartalmak összegével. A logaritmus azonosságai szerint ez is teljesül, ugyanis:

$$\sum_{j=1}^{i} \log_2 \frac{1}{p_j} = \log_2 \frac{1}{\prod_{j=1}^{i} p_j}$$

(Ez azért jó így, mert az egymástól független p_1, p_2, …, p_i események együttes bekövetkezésének valószínűsége $\prod_{j=1}^{i} p_j$.)

Ajánló

· Patkós András: Entrópia – kulcs az univerzum megismeréséhez, Természet Világa
http://www.termeszetvilaga.hu/szamok/tv2008/tv0810/patkos.html

· Wikipédia: A Shannon-féle entrópiafüggvény
http://hu.wikipedia.org/wiki/Shannon-entr%C3%B3piaf%C3%BCggv%C3%A9ny

· Játék az angol nyelv entrópiájáról
http://math.ucsd.edu/~crypto/java/ENTROPY/

Most már megfogalmazhatjuk Shannon $\sum_{i=1}^{n} p_i \cdot l_i$ minimumára vonatkozó tételét:

> Tétel:
> Ha $p_1,p_2,p_3,\dots,p_n$ valószínűségekkel bekövetkező eseményeket $l_1,l_2,l_3,\dots,l_n$ hosszúságú bináris kódszavak kódolnak – egyértelműen dekódolható módon – akkor:
>
> $$H(P) \le \min\left(\sum_{i=1}^{n} p_i \cdot l_i\right) < H(P)+1$$

A fenti tételben szerepel az egyértelműen dekódolhatóság fogalma. Ennek jelentése, hogy egy kódszavak egymás után fűzéséből kapott sorozat egyértelműen vágható szét kódszavakra, azaz ha egy üzenetben több kódszót küldünk el egymás után írva, a címzett akkor is egyértelműen kiolvashatja belőle, amit közölni akartunk. Ennek elégséges, de nem szükséges feltétele, hogy kódunk prefix legyen:

> Prefix kód:
> Nincs benne olyan kódszó, mely megegyezne egy másik kódszó elejével.

Könnyű meggondolni, hogy egy prefix kód valóban mindig egyértelműen dekódolható. Csak a szemléltetés kedvéért tegyük fel, hogy küldött üzenetünk pl. a 0110011 kódszóval kezdődik. Kódunk prefix, így ez a bitsorozat nem eleje egyetlen másik kódszónak sem, tehát az üzenet olvasója egyértelműen el tudja dönteni, hogy az első kódszó legfeljebb 7 bitből áll. Viszont kódunk prefix voltából az is következik, hogy 0, 01, 011,…,011001 bitsorozatok egyike sem lehet kódszó. Ilyen módon üzenetünk címzettje egyértelműen kiolvashatja az első kódszót. Az eljárást tovább folytatva könnyen beláthatjuk, hogy üzenetünk mindig egyértelműen dekódolható.

A tétel bizonyítása előtt lássunk néhány egyszerű példát, melyek érthetőbbé teszik az állítást!

I. Példa: Kétszer egymás után feldobunk egy pénzérmét. A következő kimenetelek lehetségesek: 2 db fejet, 2 db írást, vagy 1 fejet és 1 írást kapunk (ez utóbbi esemény kétféleképpen is bekövetkezhet, dobhatunk először fejet, utána írást, vagy fordítva, de most nem különböztetjük meg ezt a két esetet). Szeretnénk lekódolni a 2 dobás eredményét. 2 db fej, illetve 2 db írás dobásának valószínűsége $\frac{1}{4}$, 1 fej és 1 írás dobásának valószínűsége $\frac{1}{2}$ (éppen azért, mert ez a kimenetel kétféle módon is adódhat). Válasszuk a 3 szükséges kódszó hosszát a következőképpen: $l_1 = l_2 = \log_2 \frac{1}{\frac{1}{4}} = \log_2 4 = 2$, illetve $l_3 = \log_2 \frac{1}{\frac{1}{2}} = 1$. Ilyen kódszóhosszakkal találhatunk prefix, tehát egyértelműen dekódolható kódot. Legyen pl. az $\frac{1}{2}$ valószínűségű esemény kódja 1, másik két kódszavunk legyen 01 és 00. Nyilvánvaló, hogy ez a kódolás megfelel.

A fenti kód alkalmazása esetén a tétel első egyenlőtlensége egyenlőséggel teljesül:

$$\sum_{i=1}^{3} p_i \cdot l_i = 2 \cdot \frac{1}{4}\log_2 4 + \frac{1}{2}\log_2 2 = \sum_{i=1}^{3} p_i \cdot \log_2 \frac{1}{p_i} = H(P) = \frac{3}{2}$$

2. Példa

Következő példánkban $H(P) < \sum_{i=1}^{n} p_i \cdot l_i$ teljesül. Egy lezárt dobozban elhelyezünk 1 db piros, 2 db kék, 3 db zöld és 4 db sárga labdát, majd találomra kihúzunk egyet. Ekkor $\frac{1}{10}$ valószínűséggel piros, $\frac{2}{10} = \frac{1}{5}$ valószínűséggel kék, $\frac{3}{10}$ valószínűséggel zöld, végül $\frac{4}{10} = \frac{2}{5}$ valószínűséggel sárga labdát húzunk. A húzás eredményét szeretnénk lekódolni. Először próbálkozzunk minden lehetséges kimenetel esetén

a $\left\lceil \log_2 \frac{1}{p_i} \right\rceil$ kódszóhosszal (ezt a gondolatot a H(P)-t megadó egyenletben szereplő $\log_2 \frac{1}{p_i}$ kifejezés sugallhatja): $l_1 = \lceil \log_2 10 \rceil = 4$, $l_2 = \lceil \log_2 5 \rceil = 3$, $l_3 = \left\lceil \log_2 \frac{10}{3} \right\rceil = 2$, illetve $l_4 = \left\lceil \log_2 \frac{5}{2} \right\rceil = 2$. Az l_i hosszak ilyen megválasztása mellett: $\sum_{i=1}^{4} p_i \cdot l_i = \frac{1}{10} \cdot 4 + \frac{1}{5} \cdot 3 + \frac{3}{10} \cdot 2 + \frac{2}{5} \cdot 2 = 2,4$. Másrészt ezen valószínűségeloszlás entrópiája: $H(P) = \sum_{i=1}^{4} p_i \cdot \log_2 \frac{1}{p_i} = \frac{1}{10} \cdot \log_2 10 + \frac{1}{5} \cdot \log_2 5 + \frac{3}{10} \log_2 \frac{10}{3} + \frac{2}{5} \cdot \log_2 \frac{5}{2} \approx 1,846$.

Látható, hogy $H(P) < \sum_{i=1}^{4} p_i \cdot l_i < H(P) + 1$.

Néhány esetet megvizsgálva könnyen rájöhetünk, hogy a kódszavak hosszát a fentinél ügyesebben is megválaszthatjuk: $l_1 = 3$, $l_2 = 3$, $l_3 = 2$, $l_4 = 1$.
Ilyen kódszóhosszakkal készíthetünk prefix, tehát egyértelműen dekódolható kódot, pl. a következő kódszavakkal: 0, 10, 110, 111. (Persze ebből következik, hogy 2,2,3,4 kódszóhosszakkal is létezik megfelelő kód, hiszen ebben a kódban minden kódszó legalább olyan hosszú, mint az előbbi konstrukcióban.)
Ekkor:

$$\sum_{i=1}^{4} p_i \cdot l_i = \frac{1}{10} \cdot 3 + \frac{1}{5} \cdot 3 + \frac{3}{10} \cdot 2 + \frac{2}{5} \cdot 1 = 1,9$$

A $H(P) < \sum_{i=1}^{4} p_i \cdot l_i < H(P) + 1$ egyenlőtlenségek most is teljesülnek.

Térjünk vissza tételünk bizonyításához! Ehhez felhasználjuk az alábbi lemmát:

Kraft-McMillan egyenlőtlenség:

1. Ha adott egy egyértelműen dekódolható kód $l_1, l_2, \ldots, l_n$ hosszúságú kódszavakkal, akkor $\sum_{i=1}^{n} 2^{-l_i} \le 1$.
2. Ha $\sum_{i=1}^{n} 2^{-l_i} \le 1$ teljesül az $l_1, l_2, \ldots, l_n$ pozitív egészekre, akkor létezik prefix kód ezen kódszóhosszakkal.

A lemma bizonyítása előtt most is próbáljuk néhány példával érthetőbbé tenni az állítást!

3. Példa
A tétel utáni 1. példában láttuk, hogy létezik prefix kód 1, 2, 2 kódszóhosszakkal. Ezek valóban kielégítik a fenti feltételt: $2^{-1} + 2^{-2} + 2^{-2} = 1$.

4. Példa
A 2. példában 1, 2, 3, 3 kódszóhosszakra készítettünk prefix kódot: $2^{-1} + 2^{-2} + 2^{-3} + 2^{-3} = 1$.

5. Példa
Ugyanebben a példában nem használhattuk volna pl. az 1, 2, 2, 3 kódszóhosszakat, ugyanis:

$$2^{-1}+2^{-2}+2^{-2}+2^{-3}=\frac{9}{8}>1.$$

Könnyű meggondolni, hogy ilyen prefix kód valóban nem létezhet: legyen pl. az 1 bites kódszó 1. Ekkor a két db 2 bites kódszó csak 01 és 00 lehet, de ekkor nem létezik olyan 3 bites kódszó, melynek nem eleje a fenti 3 kódszó egyike sem.

$2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-6}+2^{-8}+2^{-8}<1$. Mutatunk egy prefix kódot 1,2,3,4,6,8,8 hosszúságú kódszavakkal: 1, 01, 001, 0001, 000011, 00000010, 00000001.

A példák után következhet a

lemma bizonyítása (külön látjuk be az 1. és a 2. állítást):

Az 1. állítás igazolása:

Vizsgáljuk $\left(\sum_{i=1}^{n} 2^{-l_i}\right)^k$ értékét! Jelölje C^k a k db kódszó egymás mellé írásával keletkezett kódszósorozatok halmazát. Ekkor $\left(\sum_{i=1}^{n} 2^{-l_i}\right)^k = \sum_{\sigma \in C^k} 2^{-|\sigma|}$, vagyis C^k minden σ elemének hossza megjelenik az összeg egy-egy tagjában mint 2 kitevőjének abszolút értéke. Ezt a következő gondolatmenettel láthatjuk be:

$\sum_{i=1}^{n} 2^{-l_i}$ összeget k-adik hatványra emelve, a szorzásokat elvégezve a kapott összeg minden tagja 2 egy olyan hatványa, ahol 2 kitevőjének abszolút értéke k db (persze nem feltétlenül különböző) l_j kódszóhossz összege. Tehát a kitevő abszolút értéke minden esetben C^k egy-egy elemének hossza lesz.

Másrészt az l_j hosszakból képezett minden lehetséges sorozat megjelenik az összeg egy-egy tagjában, mint 2 kitevője, így a fenti összegzést C^k összes elemére kell elvégezni, ami éppen a fenti állítás.

Jelölje $K_{l,k}$ az l hosszú C^k-beli kódszósorozatok számát. Az összes l hosszúságú σ sorozat esetén a fenti összegben 2^{-l} fog állni, és l minimális értéke $k \cdot l_{\min}$ ($l_{\min}$ az l_j kódszóhosszak közül a minimális), maximális értéke pedig $k \cdot l_{\max}$ ($l_{\max}$ az l_j kódszóhosszak maximuma), így:

$$\sum_{\sigma \in C^k} 2^{-|\sigma|} = \sum_{l=k*l_{\min}}^{k*l_{\max}} K_{l,k} \cdot 2^{-l}.$$

Most felhasználjuk, hogy kódunk egyértelműen dekódolható. Összesen 2^l db különböző l hosszúságú bináris kódszó létezik (hiszen az l db bit mindegyike 0 vagy 1 értéket vesz fel), így $K_{l,k} \le 2^l$, azaz:

$$\sum_{l=k*l_{\min}}^{k*l_{\max}} K_{l,k} \cdot 2^{-l} \le \sum_{l=k*l_{\min}}^{k*l_{\max}} 2^{-l} \cdot 2^{l} = \sum_{l=k*l_{\min}}^{k*l_{\max}} 1 \le k \cdot l_{\max}.$$

Végeredményben a következőt kaptuk:

$$\left(\sum_{i=1}^{n} 2^{-l_i}\right)^k \le k \cdot l_{\max}$$

Mindkét oldalból k-adik gyököt vonva:

$$\sum_{i=1}^{n} 2^{-l_i} \le \sqrt[k]{k \cdot l_{\max}}$$

Felhasználva, hogy $\lim_{k\to\infty} \sqrt[k]{k} = 1$ és $\lim_{k\to\infty} \sqrt[k]{l_{\max}} = 1$, a fenti egyenlőtlenség mindkét oldalának limesét véve:

$$\sum_{i=1}^{n} 2^{-l_i} \le 1.$$

Ezzel az 1. állítás bizonyítását befejeztük.

A 2. állítás igazolása:

Legyenek $l_1,l_2,...,l_n$ olyan egész számok, melyekre $\sum_{i=1}^{n} 2^{-l_i} \le 1$. Az általánosság korlátozása nélkül feltehetjük, hogy $l_1 \le l_2 \le l_3 \le ... \le l_n$. Konstruálunk egy prefix kódot ezen kódszóhosszakkal. Ehhez definiáljuk a következő $w_1, w_2, \dots, w_n$ számokat:

$$w_1 = 0$$

$$w_j = \sum_{k=1}^{j-1} 2^{l_j - l_k},\ j \in \{2,3,...,n\}.$$

Ekkor $w_j = 2^{l_j} \sum_{k=1}^{j-1} 2^{-l_k} < 2^{l_j}$, ugyanis $\sum_{k=1}^{j-1} 2^{-l_k} < \sum_{k=1}^{n} 2^{-l_k} \le 1$.

Az l_j hosszúságú kódszavunk legyen w_j 2-es számrendszerbeli alakja. $w_j < 2^{l_j}$ miatt az így kapott kódszó hosszúsága maximum l_j. Amennyiben a kódszó l_j-nél rövidebb, az elejére írt 0-kal a kívánt hosszúságúra egészíthető ki. Megmutatjuk, hogy ezzel az eljárással prefix kódhoz jutottunk. Indirekt bizonyítást alkalmazunk:

Tegyük fel, hogy a kapott kód mégsem prefix. Ekkor léteznek olyan w_i és w_j számok (i<j), hogy a w_j felhasználásával kapott kódszó eleje megegyezik a w_i felhasználásával kapott kódszóval. Két esetet vizsgálunk:

Ha $i \neq 1$, akkor w_i és w_j kettes számrendszerbeli alakja is 1-essel kezdődik, így a kód prefix volta csak úgy sérülhet, ha a két szám elé ugyanannyi 0-át írtunk, amikor a megfelelő hosszúságúra egészítettük ki őket. Ekkor w_j kettes számrendszerbeli felírásának eleje megegyezik w_i kettes számrendszerbeli alakjával. Vizsgáljuk meg egy példán, mit jelent ez:

Legyen pl. w_j=100110101 és w_i=10011. A két szám hosszának különbsége 4. w_j –t $2^4 = 16$ -tal elosztva 10011,0101-et kapunk (itt "," a "kettedesvessző"). Ennek egészrésze 10011, ami éppen w_i-vel egyenlő.

Könnyen látható, hogy a fentihez hasonló összefüggés általánosan is igaz. Képlettel megfogalmazva:

$$w_i = \left\lfloor \frac{w_j}{2^{l_j - l_i}} \right\rfloor = \left\lfloor \frac{\sum_{k=1}^{j-1} 2^{l_j - l_k}}{2^{l_j - l_i}} \right\rfloor = \left\lfloor \sum_{k=1}^{j-1} 2^{l_i - l_k} \right\rfloor$$

(Itt felhasználtuk, hogy w_j és w_i elé ugyanannyi 0-t írtunk, így l_j-l_i megegyezik a két szám hosszának különbségével.)

Másrészt a definíció alapján:

$$w_i = \sum_{k=1}^{i-1} 2^{l_i - l_k}.$$

Mivel j>i, a $\sum_{k=1}^{j-1} 2^{l_i - l_k}$ összegben szerepel $2^{l_i - l_i} = 2^0 = 1$, ez a tag a $\sum_{k=1}^{i-1} 2^{l_i - l_k}$ összegben nem jelenik meg. Másrészt a $\sum_{k=1}^{i-1} 2^{l_i - l_k}$ összeg minden tagja szerepel a $\sum_{k=1}^{j-1} 2^{l_i - l_k}$ összegben, így $\sum_{k=1}^{i-1} 2^{l_i - l_k} \le \sum_{k=1}^{j-1} 2^{l_i - l_k}$ -1, amiből $w_i = \sum_{k=1}^{i-1} 2^{l_i - l_k} < \left\lfloor \sum_{k=1}^{j-1} 2^{l_i - l_k} \right\rfloor = w_i$ következik, tehát ellentmondásra jutottunk.

Ha i=1, akkor w_i felhasználásával egy csupa 0-ból álló kódszót kapunk. Így w_j l_1 db 0-val kezdődik. Azonban a w_j-t megadó összegben szerepel $2^{l_j - l_1}$, ennek kettes számrendszerbeli alakja már önmagában $l_j - l_1 + 1$ hosszúságú, ha ez elé még l_1 db 0-t írnánk, akkor l_j-nél hosszabb kódszóhoz jutnánk, ami ismét ellentmondás.

Azzal a feltételezéssel, hogy a kapott kód nem prefix, mindkét esetben ellentmondásra jutottunk, ami a lemma helyességét bizonyítja.

Az előző bizonyításban használt konstrukciót jobban megvilágíthatjuk egy példával. A 4. példában láttuk, hogy $\sum_{i=1}^{4} 2^{-l_i} \le 1$ teljesül az l_1=1,l_2=2,l_3=3,l_4=3 kódszóhosszakra. Ekkor: $w_1 = 0$, $w_2 = 2^{l_2 - l_1} = 2$,

$w_3 = \sum_{k=1}^{2} 2^{l_3 - l_k} = 2^2 + 2^1 = 6$, és $w_4 = \sum_{k=1}^{3} 2^{l_4 - l_k} = 2^2 + 2^1 + 2^0 = 7$. E négy szám kettes számrendszerbeli alakjai rendre 0, 10, 110, 111. Látható, hogy prefix kódhoz jutottunk, megfelelő kódszóhosszakkal. (Éppen visszakaptuk a 2. példában mutatott kódszavakat.) Most nem volt szükség arra, hogy a számok kettes számrendszerbeli alakjának elejére 0 számjegyeket írjunk.

A lemma bizonyítása után hozzáláthatunk eredeti tételünk bizonyításához.

A tétel bizonyítása:

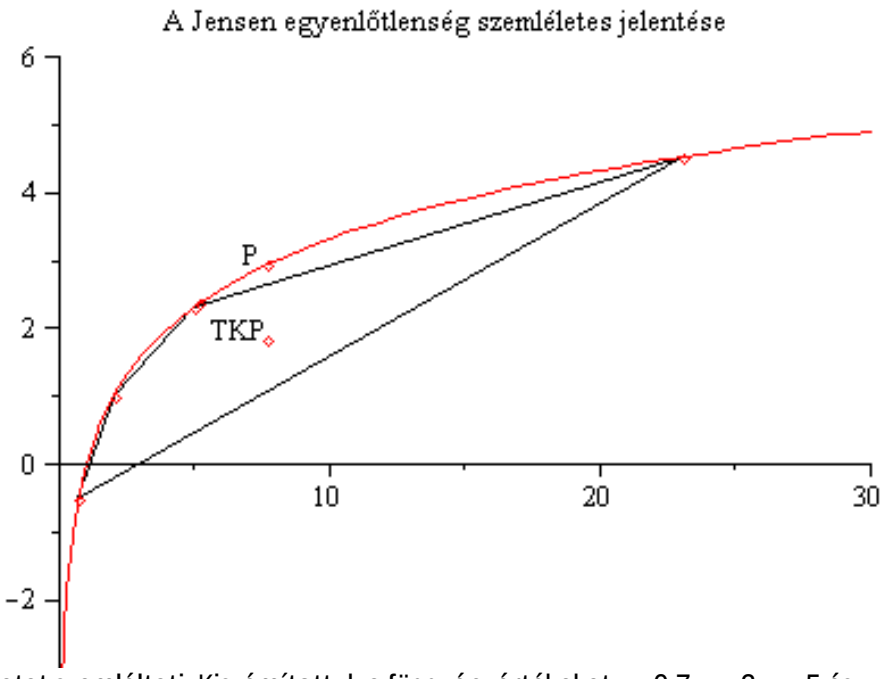
Először a $H(P) \geq \min\left(\sum_{i=1}^{n} p_i \cdot l_i\right)$ összefüggést bizonyítjuk. Vizsgáljuk a következő különbséget:

$$H(P) - \sum_{i=1}^{n} p_i \cdot l_i = \sum_{i=1}^{n} p_i \cdot \log_2 \frac{1}{p_i} + \sum_{i=1}^{n} p_i \cdot \log_2 \frac{1}{2^{l_i}} = \sum_{i=1}^{n} p_i \cdot \log_2 \frac{1}{p_i \cdot 2^{l_i}}$$

(Felhasználtuk, hogy $-l_i = \log_2 2^{-l_i} = \log_2 \frac{1}{2^{l_i}}$, és hogy az azonos alapú logaritmusok összege a szorzat logaritmusával egyezik meg.) Alkalmazzuk a Jensen egyenlőtlenséget $\sum_{i=1}^{n} p_i \cdot \log_2 \frac{1}{p_i \cdot 2^{l_i}}$-re. Eszerint a logaritmushoz hasonló konkáv függvény esetén:

$$\frac{\sum_{i=1}^{n} f(x_i)}{n} \leq f\left(\frac{\sum_{i=1}^{n} x_i}{n}\right)$$

Az egyenlőtlenség egzakt bizonyítása helyett megmutatjuk annak szemléletes jelentését:



Az ábra az n=4 esetet szemlélteti. Kiszámítottuk a függvényértékeket x_1=0,7, x_2=2, x_3=5 és x_4=23 helyeken. Az így kapott négyszög súlypontjának koordinátái:

$$\left(\frac{\sum_{i=1}^{4} x_i}{4}, \frac{\sum_{i=1}^{4} f(x_i)}{4}\right),$$

ez az ábra TKP pontja. Látható, hogy konkáv függvény esetén ez a pont a $\frac{\sum_{i=1}^{4} x_i}{4}$-hez tartozó függvényérték alá esik (ez utóbbi az ábra *P* pontja).

Alkalmazzuk a Jensen egyenlőtlenséget a logaritmusfüggvényre:

$$\sum_{i=1}^{n} p_i \cdot \log_2 \frac{1}{p_i \cdot 2^{l_i}} \le \log_2 \sum_{i=1}^{n} p_i \cdot \frac{1}{p_i \cdot 2^{l_i}} = \log_2 \sum_{i=1}^{n} 2^{-l_i} \le \log_2 1 = 0$$

Itt az utolsó egyenlőtlenségnél a Kraft-McMillan egyenlőtlenséget használtuk. Végeredményben

$H(P) - \sum_{i=1}^{n} p_i \cdot l_i \le 0$ összefüggéshez jutottunk,ami a tétel első felét bizonyítja.

Be fogjuk látni, hogy $l_i = \left\lceil \log_2 \frac{1}{p_i} \right\rceil$ kódszóhosszak esetén $\sum_{i=1}^{n} p_i \cdot l_i \le H(P)+1$ teljesül.[1] A lemma szerint ilyen kódszóhosszakkal létezik prefix (tehát egyértelműen dekódolható) kód, mivel:

$$\sum_{i=1}^{n} 2^{-\left\lceil \log_2 \frac{1}{p_i} \right\rceil} = \sum_{i=1}^{n} \frac{1}{2^{\left\lceil \log_2 \frac{1}{p_i} \right\rceil}} \le \sum_{i=1}^{n} \frac{1}{2^{\log_2 \frac{1}{p_i}}} = \sum_{i=1}^{n} p_i = 1$$

Viszont:

$$\sum_{i=1}^{n} p_i \cdot \left\lceil \log_2 \frac{1}{p_i} \right\rceil < \sum_{i=1}^{n} p_i \cdot \left(\log_2 \frac{1}{p_i} + 1 \right) = \sum_{i=1}^{n} p_i \cdot \log_2 \frac{1}{p_i} + \sum_{i=1}^{n} p_i = H(P)+1.$$

Ezzel a tételben szereplő második egyenlőtlenséget is beláttuk.

A tétel második egyenlőtlenségének bizonyításában használt $l_i = \left\lceil \log_2 \frac{1}{p_i} \right\rceil$ kódszóhosszakkal nem mindig kapunk optimális eredményt. Ez látható a Shannon tételét szemléltető 2. példából, ahol ezzel a konstrukcióval 4,3,2,2 kódszóhosszakat kapunk, pedig 3,3,2,1 hosszúságú kódszavakkal is megadható megfelelő kód.

A fenti hosszabb bizonyítás után foglalkozzunk egy kicsit azzal az úttal, melyen át üzenetünk eljut a címzetthez. Ezt a következő ábrával szemléltethetjük:

A kódoló üzenetünket a megadott kódszavak alapján kódszóvá alakítja, a dekódoló feladata felismerni, mely 0-1 sorozatot küldtük be a csatornába. Eddig csak azzal az ideális esettel foglalkoztunk, amikor üzenetünk hiba nélkül elérte a dekódolót. Ez általában nem teljesül, a fenti ábrát kiegészíthetnénk a csatornába belépő zajjal, mely üzenetünk egyes bitjeit módosíthatja. A dekódoló azért lehet képes felismerni az esetleges hibát, mert nem kerülhet be tetszőleges 0-1 sorozat a csatornába, csak azok, melyeket kódszóként kiválasztottunk. Pl. ha két kódszavunk 00 és 11, akkor a dekóder nagy valószínűséggel észreveszi, ha a csatornán való átjutás közben hiba történt (csak akkor nem, ha a 2 egymást követő bit mindegyike ellentétesre módosult). Ügyes kódolás esetén nem csak a hiba felismerése, de a hibajavítás is lehetővé válik.

Két célt kell tehát szem előtt tartanunk: alapvető elvárás, hogy üzenetünk a csatorna másik végénél is érthető legyen, az sem kívánatos azonban, hogy a küldött bitsorozat hossza túlságosan megnövekedjen. Pl. ha a rádióban a bemondó egy fontos telefonszámot csak egyszer mond el, akkor sok hallgató fogja félreérteni,

[1] A Kraft-McMillan egyenlőtlenség igazolásánál használtuk az $\lfloor x \rfloor$ jelölést, ami x alsó egészrészét, azaz a legnagyobb x-nél nem nagyobb egész számot jelentette. Most $\lceil x \rceil$ az x szám felső egészrészét, másként a legkisebb x-nél nem kisebb egész számot jelenti.

ha azonban még egyszer elismétli, akkor a hiba valószínűsége jóval kisebb lesz. Teljesen felesleges lenne viszont, ha a rádióban minden egyes mondat kétszer hangzana el, ekkor az üzenet hossza növekedne meg túlságosan. Persze a fontos információ elismétlésével még mindig nem zártuk ki a hibázás lehetőségét, a félreértés valószínűségét a lényeges adat újabb és újabb elismétlésével tovább csökkenthetjük, ekkor azonban ismét a kommunikáció sebessége csökkenne le túlságosan.

Vizsgáljunk egy *t* hosszú 0-1 sorozatot. Ha ilyen hosszúság mellett *M* db lehetséges üzenetet küldhetünk el, akkor a kommunikáció sebességét a következőképpen definiáljuk:

$$R = \frac{\log_2 M}{t}$$

Ez nyilván $M = 2^t$ esetén lenne egységnyi, ennyi különböző *t* hosszúságú üzenetet küldhetnénk, ha nem kellene a zaj miatt fellépő hibákkal foglalkozni. Ha a hibajavításra plusz biteket kell rászánnunk, akkor *R* csökken. Kódunk akkor jó, ha képesek vagyunk kis hibavalószínűség mellett is gyorsan kommunikálni. Kérdés, egy adott csatorna esetén mi az *R* sebesség szuprémuma, ha a hibázás esélyét egy előre megadott ε alá szeretnénk szorítani. (Itt a hibavalószínűséget kétféleképpen is értelmezhetjük. Egyrészt kiszámíthatjuk minden egyes kódszóra annak a valószínűségét, hogy a csatornán való átjutás után már nem (vagy rosszul) ismeri fel a dekódoló, és vehetjük ezen valószínűségek átlagát. Ezután ezt az átlagot akarjuk minimalizálni. Viszont azt is megtehetjük, hogy a különböző kódszavakra számolt hibavalószínűségek közül kiválasztjuk a maximálisat, és ennek nagyságát szeretnénk korlátozni.)

Szemléletes állítás, és hajlamosak lennénk bizonyítás nélkül elfogadni, hogy nem lehet tetszőlegesen kis hibavalószínűséggel és egyben pozitív sebességgel kommunikálni. (Fenti, rádiós bemondóval kapcsolatos példánk is azt sugallhatja, hogy az üzenet hossza a végtelenbe tart, ahogy a hiba valószínűsége 0-hoz közelít. Eközben az elküldhető üzenetek *M* száma állandó marad, a bemondó még mindig ugyannyi lehetséges telefonszám közül mond el egyet, mint korábban, *R* tehát 0-hoz tart.) Sokáig általános vélekedés volt, hogy valóban nem lehet pozitív sebességgel kommunikálni, ha a hibázás esélye 0-hoz tart. Shannon fontos felfedezése, hogy ez nem igaz: tetszőleges csatornához létezik egy küszöbszám (és ez sok fontos csatorna esetében pozitív), melynél kisebb sebességek esetén a hiba valószínűsége bármilyen kis érték alá szorítható (a küszöbszámnál nagyobb sebességek esetén pedig a hibázás valószínűsége 1-hez tart.) Ezt az éles küszöbszámot a csatorna kapacitásának nevezzük. Shannon csatornakapacitási tétele, mely az előbbi állítást kimondja, az információelmélet egyik alaptétele.

A tétel bizonyítása túlmutat ezen írás keretein. A csatornakapacitás viszont szoros kapcsolatban van egy *P* valószínűségi eloszlás *H(P)* entrópiájával, mellyel a fentiekben bővebben foglalkoztunk, ezért erről még szólunk pár szót.

Egy csatorna (legalábbis egy ún. diszkrét emlékezetnélküli csatorna, mi most csak ilyenekkel foglalkozunk) megadása a következő táblázattal lehetséges:

A táblázat minden sora elé odaírjuk a bemeneti ABC (azaz a bemeneti jelkészlet) egy-egy betűjét, oszlopainak pedig hasonlóan a kimeneti ABC egy-egy betűjét feleltetjük meg. (Az eddigiekben csak bináris kódokkal foglalkoztunk, ilyenkor az ABC a 0 és 1 jelekből áll.) Az i-edik sor j-edik oszlopába az a p valószínűség kerül, mely megmondja, az i-edik sorhoz tartozó bemenet esetén mekkora valószínűséggel kapjuk a j-edik oszlophoz írt kimenetet. Tekintsük a következő egyszerű példát:

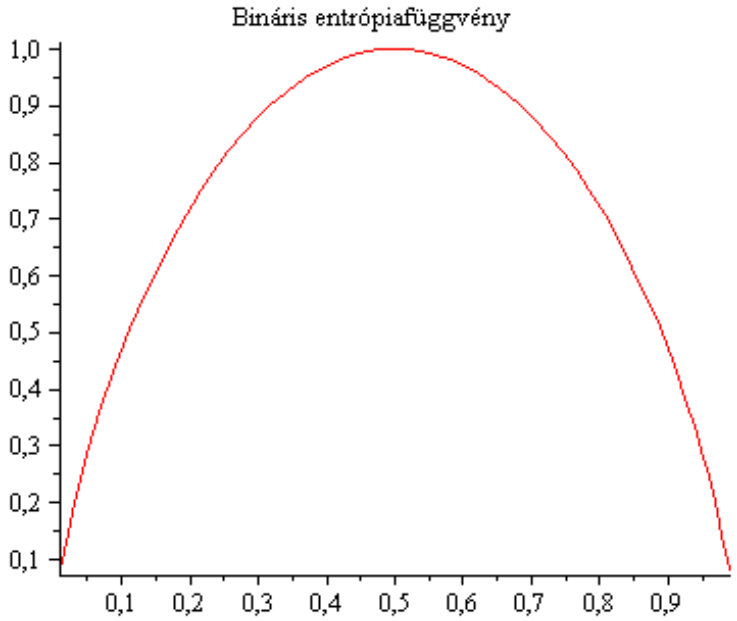
	0	1
0	1-p	p
1	p	1-p

A be- és a kimeneten egyaránt a 0 vagy az 1 bit jelenhet meg. Ha a bemeneten 0-t viszünk be, akkor a kimeneten 1-p valószínűséggel 0-t, p valószínűséggel 1-et kapunk vissza. Hasonlóan a bemeneti 1-es a kimeneten p valószínűséggel 0-t, 1-p valószínűséggel 1-est eredményez. Ideális esetben p=0, ekkor a csatornán át pluszbitek használata nélkül hibamentesen tudunk kommunikálni. Ha p elég kicsi, a hiba valószínűsége kicsi lesz. Ugyanilyen kedvező számunkra, ha p közel van 1-hez (ilyenkor a bemeneti bit nagy valószínűséggel a kimeneti bit ellentettje volt). p=0,5 esetén viszont a csatorna teljesen használhatatlan, a kimeneten kapott bit tökéletesen független a bemenettől.

Az adott csatornán át nyilván akkor tudunk jól kommunikálni, ha van néhány olyan bemenet, ami nagy valószínűséggel olyan kimenetet eredményez, ami jó közelítéssel csak az adott bemeneti jelből keletkezhet.

A csatorna be- és kimenetéhez is tartozik egy X, illetve Y, be-, illetve kimeneti ABC. A bemeneti jeleket értelmezhetjük egy értékeit X -en felvevő valószínűségi változóként.[2] (Persze X eloszlását mi határozzuk meg a kódszavak megválasztásával. Az egyes üzenetek küldésének valószínűsége a hozzájuk tartozó események bekövetkezésének valószínűségével egyezik meg, viszont mi rendeljük hozzá a kódszavakat az egyes üzenetekhez.) Ekkor a kimeneten is kapunk egy Y -on adódó valószínűségeloszlást. Kiszámíthatjuk az X valószínűségi változó eloszlásának fentebb definiált, most egyszerűen H(X)-szel jelölt entrópiáját. Ezután valamely "a" bemeneti jel esetén is kiszámolhatjuk Y entrópiáját, ezt a feltételes entrópiát H(Y|X=a)-val jelöljük. Ha a számítást az összes "a" bemenetre elvégezzük, és a kapott eredményeket átlagoljuk, akkor a H(Y|X) entrópiához jutunk, ami szemléletesen kifejezve annak a mértéke, mennyire marad bizonytalan Y, ha X-et már ismerjük. Másképpen fogalmazva: H(Y|X) megmutatja, mennyi információt hordoz Y megfigyelése még azután is, hogy X-et már tudjuk. Az Y valószínűségi változó eloszlásának is van egy H(Y) entrópiája, a H(Y)-H(Y|X) különbség tehát megadja, mennyit árul el X Y-ról. Ha ezt elfogadjuk, akkor szemléletes az az állítás, hogy nyilván ugyannyit, mint amennyit Y árul el X-ről. Tehát H(Y)-H(Y|X)=H(X)-H(X|Y). Ezt a mennyiséget az X és Y valószínűségi változók *kölcsönös információ*jának nevezzük, és I(X,Y)-nal jelöljük. A Shannon féle csatornakapacitási tételben szereplő küszöbszám (azaz a csatorna kapacitása) éppen $C=\max_{P_X} I(X,Y)$. (Nyilván az a célunk, hogy Y minél többet áruljon el X-ről. A bemenethez tartozó X eloszlást mi választhatjuk meg, ezt kell úgy variálnunk, hogy I(X,Y) maximális legyen.)

Érdemes még megemlíteni, hogy a fenti egyszerű 2x2-es táblázattal megadott csatorna esetén I(X,Y) maximális értéke az $\frac{1}{2}$- $\frac{1}{2}$ bemeneti eloszláshoz tartozik. Nyilván az a=1 és a=0 bemenet esetén is H(Y|X=a)=H(p,1-p), így az átlagolás után kapjuk: H(Y|X)=H(p,1-p). H(p,1-p)-t h(p)-vel is szokás jelölni, ennek értéke csak p-től függ. Célunk H(Y)-h(p) maximalizálása. H(Y) egy két értéket felvevő valószínűségi változó entrópiafüggvénye, az úgynevezett bináris entrópiafüggvény,ami tehát $q\times\log_2\frac{1}{q}+(1-q)\times\log_2\frac{1}{1-q}$ alakban írható, ahol q a két kimeneti jel egyike megjelenésének valószínűsége a kimeneten. A h(q) bináris entrópiafüggvény q=$\frac{1}{2}$ -nél veszi fel maximumát, ennek értéke 1, ami az ábráról is leolvasható:



Tehát I(X,Y) maximális értéke 1-h(p).

Ajánló

[2] A bemeneti ABC-t X-szel, az itt említett valószínűségi változót X-szel fogjuk jelölni. Ugyanígy különböző az Y, és az alább szintén használt Y jelölés jelentése.

· Rényi Alfréd: *Ars Mathematica* (benne: *Az információ matematikai fogalmáról*), Typotex, 2005
http://www.typotex.hu/konyv/Ars%20Mathematica
· Warren Weaver, Claude Shannon: *A kommunikáció matematikai elmélete*. Az információ elmélet születése és távlatai, Budapest, 1986, Országos Műszaki Információs Központ és Könyvtár
· Györfi László, Győri Sándor, Vajda István: *Információ és kódelmélet*
http://books.google.hu/books?id=0gfLaDnXmXEC
· Benczúr András: *Számítógépek és híradástechnika: az emberiség új kommunikációs korszaka*
http://davidalb.web.elte.hu/infkez4/Benczurjegyzet1.doc
http://davidalb.web.elte.hu/infkez4/Benczurjegyzet2.doc
· Petz Dénes: Neumann János és a kvantumbitek, előadásjegyzet
www.renyi.hu/~petz/pdf/Fazekas.pdf

A klasszikus információelmélettel való ismerkedés után térjünk át egy érdekes kapcsolat ismertetésére az információelmélet és a gráfelmélet között. Gyakorlati szempontból általában teljesen kielégítő, ha tetszőlegesen kicsi hibával tudunk kommunikálni, mégis feltehetjük a kérdést: lehetséges-e a hibátlan kommunikáció?

Térjünk vissza a csatornánkat megadó táblázathoz. A fenti egyszerű példánál $p = 0{,}1$ esetén a 0 hibával való kommunikáció nyilván teljesen reménytelen. Lássunk egy másik példát:

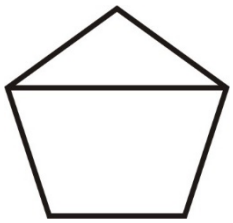
	y_1	y_2	y_3	y_4	y_5
x_1	0,1	0,2	0,4	0,2	0,1
x_2	0	0	0,7	0	0,3
x_3	0,5	0,1	0,1	0,2	0,1
x_4	0,1	0,6	0	0,3	0
x_5	0,2	0,5	0,1	0,1	0,1

Nyilvánvaló, hogy ilyen csatorna esetén az x_2 és az x_4 bemeneti jelet mindig meg tudja különböztetni egymástól a dekódoló, hiszen x_4-ből nem keletkezhet sem az y_3, sem az y_5 kimenet, x_2 bemenet esetén pedig a kimeneten biztosan ezek egyike jelenik meg. Kicsit általánosabban fogalmazva: ha két bemeneti jel olyan, hogy a táblázat bármely oszlopát vizsgálva maximum egyiküknek a sorában áll pozitív érték, akkor ezt a két jelet biztosan meg tudjuk különböztetni egymástól. Ha viszont nem létezik legalább két olyan bemeneti jel, melyek garantáltan nem téveszthetők össze, akkor biztosan nem lehetséges 0 hibával kommunikálni csatornánkon keresztül.

Azt, hogy két jelet meg lehet-e egymástól különböztetni, könnyen leírhatjuk a következő gráffal: a gráf csúcsai legyenek a bemeneti jelek, két csúcs között akkor fusson él, ha a csúcsokhoz rendelt jelek biztosan nem téveszthetők össze. Az így kapott gráfot a csatorna bemeneti betűihez tartozó megkülönböztethetőségi gráfnak nevezzük (az összetéveszthetőségi gráf, mely ennek komplementere, teljesen hasonlóan definiálható). Kérdés, egy adott G megkülönböztethetőségi gráfú csatorna esetén hány olyan t hosszúságú kódszót készíthetünk, melyek páronként megkülönböztethetők. Ha M a megfelelő kódszavak lehetséges maximális száma, akkor ismét a $\frac{\log_2 M}{t}$ hányados határértékét keressük $t \to \infty$ esetén. Ezt a számot a csatorna zéróhiba-kapacitásának nevezzük.

Ennek vizsgálatához érdemes elkészíteni a G gráf megfelelőjét a bemeneti jelekből alkotott t hosszú sorozatokra, ezt G^t-vel jelöljük. G^t csúcsai az egyes t hosszú sorozatok, két csúcs között akkor fut él, ha a hozzájuk rendelt sorozatok egymástól megkülönböztethetők. Ez akkor következik be, ha a két sorozat legalább egyetlen helyen nem összetéveszthető, azaz legalább egy helyen a G gráf élét alkotta.

Egy F gráf fontos jellemzője az $\omega(F)$ klikkszám, amely F legnagyobb teljes részgráfjának mérete. (Teljes gráfról akkor beszélünk, ha minden csúcs az összes többi csúccsal össze van kötve.) Nyilván maximum annyi páronként megkülönböztethető t hosszúságú sorozatot készíthetünk, amennyi G^t klikkszáma. Pl. tekintsük a következő megkülönböztethetőségi gráfhoz tartozó csatornát:



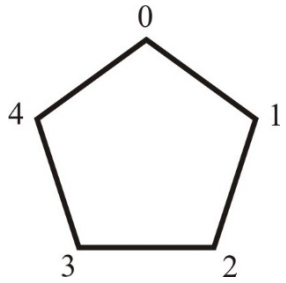
Itt G legnagyobb teljes részgráfja egy háromszög, $\omega(G)=3$, így maximum 3 páronként megkülönböztethető 1 hosszú üzenetet küldhetünk csatornánkon át. Ahhoz, hogy a páronként megkülönböztethető t hosszú sorozatok számát meghatározzuk, az $\omega(G^t)$ klikkszámot kellene ismernünk.

Eredeti problémánkat így tisztán gráfelméleti feladattá fogalmazhatjuk át: a G gráf Shannon kapacitását akarjuk meghatározni, melyet a következőképpen definiálunk[3]:

$$C(G)=\lim_{t\to\infty}\frac{\log_2 \omega(G^t)}{t}.$$

A Shannon kapacitás olyan gráfparaméter, melynek kiszámítása komoly nehézségekbe ütközik. Nem egyszerűen arról van szó, hogy egy nagy méretű gráf esetén nehéz algoritmust találni a meghatározására (sok gráfparaméter esetén ez a helyzet), az sem ismert, hogy egyáltalán létezik-e megfelelő algoritmus. Már egészen kis csúcsszámú gráfoknál előfordulnak olyan esetek, amikor a gráf Shannon kapacitását nem ismerjük.

A G gráf Shannon kapacitásának nyilvánvaló alsó korlátja $\log_2 \omega(G)$, ugyanis ha üzeneteinket csak azokból a bemeneti jelekből állítjuk elő, melyek páronként megkülönböztethetőek, akkor nyilván páronként össze nem téveszthető t hosszú sorozatokat kapunk. Ilyen sorozatokból éppen $(\omega(G))^t$ készíthető, így valóban $C(G)=\lim_{t\to\infty}\frac{\log_2 \omega(G^t)}{t} \geq \lim_{t\to\infty}\frac{\log_2(\omega(G))^t}{t}=\log_2 \omega(G)$. Kérdés, fölé lehet-e menni ennek az alsó korlátnak. A válasz igen, a legkisebb csúcsszámú olyan gráf, melyben a fenti kifejezésben szigorú egyenlőtlenség áll, az 5 hosszúságú kör:



Ennek a gráfnak a klikkszáma 2, így ha csak páronként megkülönböztethető bemeneti jeleket használunk a 2 hosszú sorozatokban, akkor csak $2^2=4$ sorozatot készíthetünk. Ügyesebb módszerrel 5 páronként össze nem téveszthető 2 hosszúságú sorozatot is megadhatunk:

00, 12, 24, 31, 43.

A felsorolásban egymás után következő üzenetek (valamint az utolsó és az első üzenet) első betűi össze nem téveszthetők, a felsorolásban másodszomszédos sorozatoknak (illetve a negyedik és az első, valamint az utolsó és a második sorozatnak) pedig a második betűje megkülönböztethető, így az összes sorozat páronként megkülönböztethető. Ha $t=2\times k$, akkor ezt az öt 2 hosszúságú sorozatot felhasználva 5^k darab t hosszú sorozatot készíthetünk, melyek páronként megkülönböztethetőek. Ebből azonnal adódik, hogy az 5 hosszú kör kapacitása legalább $\sqrt{5}$. Sokáig nyitott kérdés maradt, ez-e az optimális érték. Lovász László egyik híres eredménye annak a bizonyítása, hogy az 5 hosszúságú kör Shannon kapacitása valóban $\sqrt{5}$-tel egyenlő. (Ennek jelentősége abban is áll, hogy a bizonyítás során Lovász bevezetett egy később nagyon fontossá vált gráfparamétert.)

[3] Megjegyezzük, hogy sok tárgyalás az összetéveszthetőségi gráf segítségével, ezért komplementer módon definiálja ezt a fogalmat, vagyis úgy, hogy amit mi C(G)-vel jelölünk, azt ezen tárgyalások a komplementer gráf Shannon kapacitásának nevezik.

Ajánló

· Lovász László: On the Shannon capacity of a graph
http://www.cs.elte.hu/~lovasz/scans/theta.pdf

· Tom Bohman, Ron Holzman: A nontrivial lower bound on the Shannon capacities of the complements of odd cycles
http://www2.technion.ac.il/~holzman/papers/completr.pdf

· Noga Alon: The Shannon capacity of a union
http://www.math.tau.ac.il/~nogaa/PDFS/shann3.pdf

Megmutattuk, hogy egy G gráf Shannon kapacitása alulról becsülhető $\log_2 w(G)$-vel. Könnyen bebizonyítható, hogy C(G)-re felső korlát lesz $\log_2 c(G)$, a gráf kromatikus számának logaritmusa.

Egy gráf kromatikus száma azt adja meg, legalább hány színre van szükség a gráf csúcsainak olyan színezéséhez, ahol az egymással szomszédos csúcsok (azok a csúcsok, melyek között fut él) különböző színűek. Nyilvánvaló, hogy $w(G) \le c(G)$ mindig teljesül, hiszen a gráfban van $w(G)$ darab páronként összekötött csúcs, melyek közül semelyik kettő nem lehet azonos színű. Mielőtt bebizonyítanánk, hogy $C(G) \le \log_2 c(G)$, lássuk be a következő segédtételt:

Lemma:
$$c(G^t) \le (c(G))^t$$

A lemma bizonyítása:
Vegyük G egy jó színezését $c(G)$ színnel. Ennek segítségével fogjuk definiálni a G^t gráf egy megfelelő színezését $(c(G))^t$ színnel. G^t minden csúcsa egy-egy G csúcsaiból álló sorozat. Egy ilyen csúcsot színezzünk a G-beli megfelelő színek sorozatával. Ha két G^t –beli csúcs össze van kötve, akkor valamely i-re a csúcsok i-edik koordinátájában G-ben összekötött csúcsok állnak, és így ezek színe különböző az eredeti gráf színezésében. Tehát itt a színsorozat is eltér, vagyis G^t –ben az összekötött csúcsokat különböző színekkel színeztük. A $c(G)$ színből legfeljebb $(c(G))^t$ különböző t hosszúságú színsorozat készíthető, tehát legfeljebb ennyi színt használtunk fel G^t jó színezéséhez. Így G^t legfeljebb $(c(G))^t$ színnel kiszínezhető, kromatikus száma nem lehet ennél nagyobb. Ezzel a lemma állítását beláttuk.

Ezután lássuk a tétel bizonyítását:

Tétel:
$$C(G) \le \log_2 c(G)$$

Bizonyítás:
$$w(G^t) \le c(G^t) \le (c(G))^t$$
Vonjunk t-edik gyököt az egyenlőtlenség két oldalából, és vegyük a kettes alapú logaritmust:
$$\log_2(w(G^t)) \le t \cdot \log_2 c(G)$$
Osszunk t-vel, és vizsgáljuk mindkét oldal határértékét $t \to \infty$ esetén. A bal oldalon definíció szerint C(G) fog állni, így valóban:
$$C(G) \le \log_2 c(G).$$

Végeredményben a következő egyenlőtlenséglánchoz jutottunk:
$$\log_2 w(G) \le C(G) \le \log_2 c(G).$$

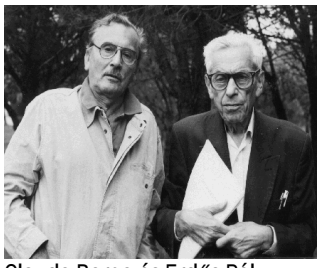
Ha $w(G) = c(G)$, ez lehetőséget nyújt arra, hogy C(G)-t meghatározzuk, de sok gráf esetén ez az egyenlőség nem áll. A legegyszerűbb olyan gráf, ahol nem teljesül az egyenlőség, éppen a fenti példánál látott 5 hosszúságú kör.

Vajon milyenek lehetnek azok a gráfok, melyekre $w(G) = c(G)$ teljesül? Ezzel a kérdéssel az 1960-as évek elején kezdett el foglalkozni Claude Berge francia matematikus, éppen a Shannon-féle gráfkapacitás fenti tulajdonsága által inspirálva. Az, hogy egy G gráfnál ez a két gráfparaméter egyenlő, még nem mond sokat a gráf struktúrájáról, hiszen ha tekintünk egy olyan gráfot, ahol ez a két érték nagyon eltér egymástól, majd mellérakunk egy elég nagy teljes gráfot, akkor az így kapott G gráfban $c(G) = w(G)$ teljesülni fog. Ezért Berge a következő kikötést tette: vegyük azokat a gráfokat, amelyek esetén mind az eredeti gráfra, mind annak összes feszített részgráfjára fennáll az $c(G) = w(G)$ egyenlőség (feszített részgráfhoz akkor jutunk, ha a gráf bizonyos pontjait, valamint valamennyi ezen csúcsok között futó élt kijelöljük, a többi csúcsot és élt pedig elhagyjuk). Az ilyen tulajdonságú gráfokat Berge perfekt gráfoknak nevezte el. Az évek során kiderült, hogy ezek igen fontos, sok érdekes tulajdonsággal rendelkező gráfosztályt alkotnak.

Berge megfogalmazta azt a sejtést, hogy ha egy gráf perfekt, akkor a komplementere is az. Lovász egy másik híres eredménye ennek a sejtésnek a bizonyítása. Egy az előzőnél erősebb, szintén Berge-től származó sejtés évtizedekig nyitva maradt, több mint 150 oldalas cikkben megjelent bizonyítása csak pár éve született meg: pontosan azok a gráfok perfektek, melyek esetén sem a gráf, sem a komplementere nem tartalmaz feszített részgráfként (vagyis 'átlói' nélkül) legalább 5 hosszúságú páratlan kört. Ezen tétel bizonyítása (melyet Maria Chudnovsky, Neil Robertson, Paul Seymour és Robin Thomas talált meg) az utóbbi évek egyik legnagyobb gráfelméleti eredménye.

Láttuk, hogy az információelmélet nem csak önmagában jelentős ága a matematikának, hanem más területekre is inspirálóan hatott. A Shannon által bevezetett fogalmak az eredeti problémától igen messzire vezettek, és például a gráfelméletben is számos új, érdekes eredmény megszületését inspirálták.

Ajánló



Claude Berge és Erdős Pál
Chronomaths
http://serge.mehl.free.fr/

· A 2004. évi párizsi Claude Berge emlékülés oldala
http://www.ecp6.jussieu.fr/GT04/Berge/Berge.html
· Vašek Chvátal: Claude Berge 5. 6. 1926 – 30. 6. 2002
http://users.encs.concordia.ca/~chvatal/perfect/claude2.pdf
· Denis Boyssou, Dominique de Werra, Olivier Hudry: Claude Berge and the „Oulipo"
http://www.lamsade.dauphine.fr/~bouyssou/Berge.pdf
· Recski András: Gráfok színezése
http://matek.fazekas.hu/portal/eloadas/2007/eloadas_2008_01_22_recski.html
· Wikipédia a perfekt gráfokról:
http://en.wikipedia.org/wiki/Perfect_graph
· M. Chudnovsky, N. Robertson, P. D. Seymour, R. Thomas: Progress on perfect graphs,
http://people.math.gatech.edu/~thomas/PAP/perfsur.pdf
· Paul Seymour: How the proof of the strong perfect graph conjecture was found
http://www.math.princeton.edu/~pds/papers/howtheperfect/howtheperfect.pdf
· Tony Jebara előadása a Perfekt gráfokról:
http://videolectures.net/mlss09us_jebara_mapepg/
· Vašek Chvátal: Perfect problems
http://users.encs.concordia.ca/~chvatal/perfect/problems.html
· Simonyi Gábor: Graph Entropy – A Survey
www.renyi.hu/~simonyi/grams.ps

Simonyi Gábor
Információközlés és gráfelmélet

A 2009. szeptember 29-i előadás kibővített változata
Lejegyezte és szerkesztette Lovas Lia Izabella

Bevezető feladat: Valaki kiválasztja egy sakktábla egy tetszőleges mezőjét. Legalább hány eldöntendő kérdésre van szükségünk ahhoz, hogy biztosan kitalálhassuk a gondolt mezőt?

A feladat megoldása igen egyszerű: 6 kérdés elég, ugyanis minden lépésben feloszthatjuk két egyenlő részre a még szóba jöhető mezőket, és rákérdezhetünk, hogy a keresett mező melyik csoportban található. Ilyen módon a hatodik kérdés után egyetlen mező marad. Másrészt hatnál kevesebb kérdés nem lehet elég: ha a még ki nem zárt mezőket következő kérdésünk két nem egyenlő csoportra bontja, akkor mindig lehetséges, hogy a kiválasztott mező a nagyobb elemszámú halmazba kerül.

Felvetődik: vajon akkor is 6-e a fenti feladat megoldása, ha előre meg kell adnunk az összes kérdésünket, és csak ezután kapjuk meg a válaszokat?

Megoldás: Igen. Képzeljük el, hogy minden mezőhöz hozzárendelünk egy 6 karakterből álló 0-1 sorozatot. Ezekből éppen $2^6 = 64$ különböző létezik, így a sakktábla minden mezejéhez különböző sorozatot rendelhetünk. Első kérdésünk az lehet, 1-es-e a mezőhöz tartozó sorozat első eleme, majd ugyanígy végigmehetünk a sorozat összes bitjén. A hatodik kérdés után ismerni fogjuk a mezőhöz rendelt teljes sorozatot, tehát magát a mezőt is kitaláltuk.

A fenti egyszerű példában a sakktábla mezőihez 0-1 sorozatokat rendeltünk, lényegében kódoltuk őket. Egy-egy ilyen 0-kból és 1-esekből álló sorozatot a későbbiekben *bináris kódszó*nak fogunk nevezni.

Az információelmélet születése Claude Shannon nevéhez fűződik, aki 1948-ban megjelent A Mathematical Theory of Communication című munkájában lefektette a matematika ezen új területének alapjait. A későbbi évek jelentős eredményei közül is számos az ő nevéhez fűződik. Ezen írás keretein belül csak arra van lehetőségünk, hogy rövid ízelítőt adjunk az információelmélet alapfogalmaiból, illetve vázlatosan rávilágítsunk egy érdekes kapcsolatra a gráfelmélettel.

Ajánló
· A Mathematical Theory of Communication:
http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html
· Aaron D. Wyner: Shannon művének jelentősége:
http://cm.bell-labs.com/cm/ms/what/shannonday/work.html
· Claude Shannon a MacTutor Matematikatörténeti Gyűjteményben:
http://www-groups.dcs.st-and.ac.uk/~history/Biographies/Shannon.html
· Claude Shannon – Father of the information Age (film):
http://www.youtube.com/watch?v=z2Whj_nL-x8

Egy adott üzenet információtartalmát a kódolásához minimálisan szükséges bitek számával fogjuk mérni. Érdemes megfigyelni, hogy ez teljesen független az üzenet tartalmától, amitől függ, az a lehetséges üzenetek száma. Adott kommunikációs helyzetben törekszünk arra, hogy minél rövidebb üzenetet küldjünk. Pontosabban fogalmazva, azt akarjuk elérni, hogy üzenetünk várható hossza minimális legyen. Figyelembe szeretnénk venni, hogy egy esemény lehetséges kimenetelei közül nem biztos, hogy mindegyik azonos valószínűséggel következik be. Ilyenkor nem biztos, hogy minden kimenetelt érdemes azonos hosszúságú kódszavakkal kódolni. Vegyünk egy példát:

Minden héten lottózunk, és hónap végén (azaz négyhetenként; az egyszerűség kedvéért feltesszük, hogy minden hónap 4 hétből áll) szeretnénk egy üzenetben elküldeni, melyik héten nyertünk, és melyiken nem. Ez nyilván megoldható, ha 4 hosszúságú bináris kódot alkalmazunk: azokhoz a hetekhez, amikor nem nyertünk, 0-t rendelünk, ellenkező esetben 1-et. Ez a módszer nem túl gazdaságos: ha az eljárást minden hónap végén megismételjük, az esetek döntő többségében a 0000 sorozatot fogjuk elküldeni. Ehelyett küldhetünk pl. egyetlen 0 bitet, a többi, nagyon kis valószínűségű esetet pedig 1 bitnél hosszabb sorozatokkal kódoljuk. Ha az üzenetek küldését hosszú időn át folytatjuk, átlagosan nyilván 4-nél jóval kevesebb bitet kell elküldenünk havonta.

A továbbiakban is bináris kódokkal foglalkozunk. Vizsgáljuk azt az esetet, amikor n lehetséges kimenetel van, az i-edik kimenetel valószínűsége p_i, a hozzá rendelt kódszó hossza pedig l_i. Célunk, hogy a $\sum_{i=1}^{n} p_i \cdot l_i$ összeget, ami a küldött üzenet átlagos hossza (másként fogalmazva: az üzenet hosszának várható értéke), minimalizáljuk.

Meg fogjuk mutatni, hogy $\sum_{i=1}^{n} p_i \cdot l_i$ alulról becsülhető a $P=(p_1,p_2,p_3,\dots,p_n)$ valószínűségeloszlás entrópiájával, melyet az alábbi módon definiálunk (p_i=0 esetén a megfelelő tagot 0-nak vesszük):

Entrópia:

$$H(P)=\sum_{i=1}^{n} p_i \cdot \log_2 \frac{1}{p_i}$$

Szokás ezt úgy értelmezni, hogy a p_i valószínűségű esemény bekövetkezésének információtartalma $\log_2 \frac{1}{p_i}$, és így az adott valószínűségi változó értékei átlagosan $H(P)$ információt hordoznak. Az alábbi egyszerű tények azt mutatják, hogy ez az értelmezés összhangban van néhány természetes elvárással:

- Biztos esemény bekövetkezése nem ad információt, így elvárjuk, hogy a p_i=1-hez tartozó esemény információtartalma 0 legyen. $\log_2 1 = 0$ valóban teljesül.
- Két egyforma valószínűségű esemény egyikének bekövetkezte jelentsen 1 bit információt. Ennek a feltételnek is megfelel a fenti értelmezés: $\log_2 \frac{1}{0{,}5} = 1$.
- Egymástól független események együttes bekövetkezésének információtartalma egyezzen meg az egyes események bekövetkezte által hordozott információtartalmak összegével. A logaritmus azonosságai szerint ez is teljesül, ugyanis:

$$\sum_{j=1}^{i} \log_2 \frac{1}{p_j} = \log_2 \frac{1}{\prod_{j=1}^{i} p_j}$$

(Ez azért jó így, mert az egymástól független p_1, p_2, …, p_i események együttes bekövetkezésének valószínűsége $\prod_{j=1}^{i} p_j$.)

Ajánló

· Patkós András: Entrópia – kulcs az univerzum megismeréséhez, Természet Világa
http://www.termeszetvilaga.hu/szamok/tv2008/tv0810/patkos.html

· Wikipédia: A Shannon-féle entrópiafüggvény
http://hu.wikipedia.org/wiki/Shannon-entr%C3%B3piaf%C3%BCggv%C3%A9ny

· Játék az angol nyelv entrópiájáról
http://math.ucsd.edu/~crypto/java/ENTROPY/

Most már megfogalmazhatjuk Shannon $\sum_{i=1}^{n} p_i \cdot l_i$ minimumára vonatkozó tételét:

> Tétel:
> Ha $p_1,p_2,p_3,\dots,p_n$ valószínűségekkel bekövetkező eseményeket $l_1,l_2,l_3,\dots,l_n$ hosszúságú bináris kódszavak kódolnak – egyértelműen dekódolható módon – akkor:
>
> $$H(P) \le \min\left(\sum_{i=1}^{n} p_i \cdot l_i\right) < H(P)+1$$

A fenti tételben szerepel az egyértelműen dekódolhatóság fogalma. Ennek jelentése, hogy egy kódszavak egymás után fűzéséből kapott sorozat egyértelműen vágható szét kódszavakra, azaz ha egy üzenetben több kódszót küldünk el egymás után írva, a címzett akkor is egyértelműen kiolvashatja belőle, amit közölni akartunk. Ennek elégséges, de nem szükséges feltétele, hogy kódunk prefix legyen:

> Prefix kód:
> Nincs benne olyan kódszó, mely megegyezne egy másik kódszó elejével.

Könnyű meggondolni, hogy egy prefix kód valóban mindig egyértelműen dekódolható. Csak a szemléltetés kedvéért tegyük fel, hogy küldött üzenetünk pl. a 0110011 kódszóval kezdődik. Kódunk prefix, így ez a bitsorozat nem eleje egyetlen másik kódszónak sem, tehát az üzenet olvasója egyértelműen el tudja dönteni, hogy az első kódszó legfeljebb 7 bitből áll. Viszont kódunk prefix voltából az is következik, hogy 0, 01, 011,…,011001 bitsorozatok egyike sem lehet kódszó. Ilyen módon üzenetünk címzettje egyértelműen kiolvashatja az első kódszót. Az eljárást tovább folytatva könnyen beláthatjuk, hogy üzenetünk mindig egyértelműen dekódolható.

A tétel bizonyítása előtt lássunk néhány egyszerű példát, melyek érthetőbbé teszik az állítást!

I. Példa: Kétszer egymás után feldobunk egy pénzérmét. A következő kimenetelek lehetségesek: 2 db fejet, 2 db írást, vagy 1 fejet és 1 írást kapunk (ez utóbbi esemény kétféleképpen is bekövetkezhet, dobhatunk először fejet, utána írást, vagy fordítva, de most nem különböztetjük meg ezt a két esetet). Szeretnénk lekódolni a 2 dobás eredményét. 2 db fej, illetve 2 db írás dobásának valószínűsége $\frac{1}{4}$, 1 fej és 1 írás dobásának valószínűsége $\frac{1}{2}$ (éppen azért, mert ez a kimenetel kétféle módon is adódhat). Válasszuk a 3 szükséges kódszó hosszát a következőképpen: $l_1 = l_2 = \log_2 \frac{1}{\frac{1}{4}} = \log_2 4 = 2$, illetve $l_3 = \log_2 \frac{1}{\frac{1}{2}} = 1$. Ilyen kódszóhosszakkal találhatunk prefix, tehát egyértelműen dekódolható kódot. Legyen pl. az $\frac{1}{2}$ valószínűségű esemény kódja 1, másik két kódszavunk legyen 01 és 00. Nyilvánvaló, hogy ez a kódolás megfelel.

A fenti kód alkalmazása esetén a tétel első egyenlőtlensége egyenlőséggel teljesül:

$$\sum_{i=1}^{3} p_i \cdot l_i = 2 \cdot \frac{1}{4}\log_2 4 + \frac{1}{2}\log_2 2 = \sum_{i=1}^{3} p_i \cdot \log_2 \frac{1}{p_i} = H(P) = \frac{3}{2}$$

2. Példa

Következő példánkban $H(P) < \sum_{i=1}^{n} p_i \cdot l_i$ teljesül. Egy lezárt dobozban elhelyezünk 1 db piros, 2 db kék, 3 db zöld és 4 db sárga labdát, majd találomra kihúzunk egyet. Ekkor $\frac{1}{10}$ valószínűséggel piros, $\frac{2}{10} = \frac{1}{5}$ valószínűséggel kék, $\frac{3}{10}$ valószínűséggel zöld, végül $\frac{4}{10} = \frac{2}{5}$ valószínűséggel sárga labdát húzunk. A húzás eredményét szeretnénk lekódolni. Először próbálkozzunk minden lehetséges kimenetel esetén

a $\left\lceil \log_2 \frac{1}{p_i} \right\rceil$ kódszóhosszal (ezt a gondolatot a H(P)-t megadó egyenletben szereplő $\log_2 \frac{1}{p_i}$ kifejezés sugallhatja): $l_1 = \lceil \log_2 10 \rceil = 4$, $l_2 = \lceil \log_2 5 \rceil = 3$, $l_3 = \left\lceil \log_2 \frac{10}{3} \right\rceil = 2$, illetve $l_4 = \left\lceil \log_2 \frac{5}{2} \right\rceil = 2$. Az l_i hosszak ilyen megválasztása mellett: $\sum_{i=1}^{4} p_i \cdot l_i = \frac{1}{10} \cdot 4 + \frac{1}{5} \cdot 3 + \frac{3}{10} \cdot 2 + \frac{2}{5} \cdot 2 = 2,4$. Másrészt ezen valószínűségeloszlás entrópiája: $H(P) = \sum_{i=1}^{4} p_i \cdot \log_2 \frac{1}{p_i} = \frac{1}{10} \cdot \log_2 10 + \frac{1}{5} \cdot \log_2 5 + \frac{3}{10} \log_2 \frac{10}{3} + \frac{2}{5} \cdot \log_2 \frac{5}{2} \approx 1,846$.

Látható, hogy $H(P) < \sum_{i=1}^{4} p_i \cdot l_i < H(P) + 1$.

Néhány esetet megvizsgálva könnyen rájöhetünk, hogy a kódszavak hosszát a fentinél ügyesebben is megválaszthatjuk: $l_1 = 3$, $l_2 = 3$, $l_3 = 2$, $l_4 = 1$.
Ilyen kódszóhosszakkal készíthetünk prefix, tehát egyértelműen dekódolható kódot, pl. a következő kódszavakkal: 0, 10, 110, 111. (Persze ebből következik, hogy 2,2,3,4 kódszóhosszakkal is létezik megfelelő kód, hiszen ebben a kódban minden kódszó legalább olyan hosszú, mint az előbbi konstrukcióban.)
Ekkor:

$$\sum_{i=1}^{4} p_i \cdot l_i = \frac{1}{10} \cdot 3 + \frac{1}{5} \cdot 3 + \frac{3}{10} \cdot 2 + \frac{2}{5} \cdot 1 = 1,9$$

A $H(P) < \sum_{i=1}^{4} p_i \cdot l_i < H(P) + 1$ egyenlőtlenségek most is teljesülnek.

Térjünk vissza tételünk bizonyításához! Ehhez felhasználjuk az alábbi lemmát:

Kraft-McMillan egyenlőtlenség:

1. Ha adott egy egyértelműen dekódolható kód $l_1, l_2, \ldots, l_n$ hosszúságú kódszavakkal, akkor $\sum_{i=1}^{n} 2^{-l_i} \le 1$.
2. Ha $\sum_{i=1}^{n} 2^{-l_i} \le 1$ teljesül az $l_1, l_2, \ldots, l_n$ pozitív egészekre, akkor létezik prefix kód ezen kódszóhosszakkal.

A lemma bizonyítása előtt most is próbáljuk néhány példával érthetőbbé tenni az állítást!

3. Példa
A tétel utáni 1. példában láttuk, hogy létezik prefix kód 1, 2, 2 kódszóhosszakkal. Ezek valóban kielégítik a fenti feltételt: $2^{-1} + 2^{-2} + 2^{-2} = 1$.

4. Példa
A 2. példában 1, 2, 3, 3 kódszóhosszakra készítettünk prefix kódot: $2^{-1} + 2^{-2} + 2^{-3} + 2^{-3} = 1$.

5. Példa
Ugyanebben a példában nem használhattuk volna pl. az 1, 2, 2, 3 kódszóhosszakat, ugyanis:

$$2^{-1}+2^{-2}+2^{-2}+2^{-3}=\frac{9}{8}>1.$$

Könnyű meggondolni, hogy ilyen prefix kód valóban nem létezhet: legyen pl. az 1 bites kódszó 1. Ekkor a két db 2 bites kódszó csak 01 és 00 lehet, de ekkor nem létezik olyan 3 bites kódszó, melynek nem eleje a fenti 3 kódszó egyike sem.

$2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-6}+2^{-8}+2^{-8}<1$. Mutatunk egy prefix kódot 1,2,3,4,6,8,8 hosszúságú kódszavakkal: 1, 01, 001, 0001, 000011, 00000010, 00000001.

A példák után következhet a

lemma bizonyítása (külön látjuk be az 1. és a 2. állítást):

Az 1. állítás igazolása:

Vizsgáljuk $\left(\sum_{i=1}^{n} 2^{-l_i}\right)^k$ értékét! Jelölje C^k a k db kódszó egymás mellé írásával keletkezett kódszósorozatok halmazát. Ekkor $\left(\sum_{i=1}^{n} 2^{-l_i}\right)^k = \sum_{\sigma \in C^k} 2^{-|\sigma|}$, vagyis C^k minden σ elemének hossza megjelenik az összeg egy-egy tagjában mint 2 kitevőjének abszolút értéke. Ezt a következő gondolatmenettel láthatjuk be:

$\sum_{i=1}^{n} 2^{-l_i}$ összeget k-adik hatványra emelve, a szorzásokat elvégezve a kapott összeg minden tagja 2 egy olyan hatványa, ahol 2 kitevőjének abszolút értéke k db (persze nem feltétlenül különböző) l_j kódszóhossz összege. Tehát a kitevő abszolút értéke minden esetben C^k egy-egy elemének hossza lesz.

Másrészt az l_j hosszakból képezett minden lehetséges sorozat megjelenik az összeg egy-egy tagjában, mint 2 kitevője, így a fenti összegzést C^k összes elemére kell elvégezni, ami éppen a fenti állítás.

Jelölje $K_{l,k}$ az l hosszú C^k-beli kódszósorozatok számát. Az összes l hosszúságú σ sorozat esetén a fenti összegben 2^{-l} fog állni, és l minimális értéke $k \cdot l_{\min}$ ($l_{\min}$ az l_j kódszóhosszak közül a minimális), maximális értéke pedig $k \cdot l_{\max}$ ($l_{\max}$ az l_j kódszóhosszak maximuma), így:

$$\sum_{\sigma \in C^k} 2^{-|\sigma|} = \sum_{l=k*l_{\min}}^{k*l_{\max}} K_{l,k} \cdot 2^{-l}.$$

Most felhasználjuk, hogy kódunk egyértelműen dekódolható. Összesen 2^l db különböző l hosszúságú bináris kódszó létezik (hiszen az l db bit mindegyike 0 vagy 1 értéket vesz fel), így $K_{l,k} \le 2^l$, azaz:

$$\sum_{l=k*l_{\min}}^{k*l_{\max}} K_{l,k} \cdot 2^{-l} \le \sum_{l=k*l_{\min}}^{k*l_{\max}} 2^{-l} \cdot 2^{l} = \sum_{l=k*l_{\min}}^{k*l_{\max}} 1 \le k \cdot l_{\max}.$$

Végeredményben a következőt kaptuk:

$$\left(\sum_{i=1}^{n} 2^{-l_i}\right)^k \le k \cdot l_{\max}$$

Mindkét oldalból k-adik gyököt vonva:

$$\sum_{i=1}^{n} 2^{-l_i} \le \sqrt[k]{k \cdot l_{\max}}$$

Felhasználva, hogy $\lim_{k\to\infty} \sqrt[k]{k} = 1$ és $\lim_{k\to\infty} \sqrt[k]{l_{\max}} = 1$, a fenti egyenlőtlenség mindkét oldalának limesét véve:

$$\sum_{i=1}^{n} 2^{-l_i} \le 1.$$

Ezzel az 1. állítás bizonyítását befejeztük.

A 2. állítás igazolása:

Legyenek $l_1,l_2,...,l_n$ olyan egész számok, melyekre $\sum_{i=1}^{n} 2^{-l_i} \le 1$. Az általánosság korlátozása nélkül feltehetjük, hogy $l_1 \le l_2 \le l_3 \le ... \le l_n$. Konstruálunk egy prefix kódot ezen kódszóhosszakkal. Ehhez definiáljuk a következő $w_1, w_2, \dots, w_n$ számokat:

$$w_1 = 0$$

$$w_j = \sum_{k=1}^{j-1} 2^{l_j - l_k},\ j \in \{2,3,...,n\}.$$

Ekkor $w_j = 2^{l_j} \sum_{k=1}^{j-1} 2^{-l_k} < 2^{l_j}$, ugyanis $\sum_{k=1}^{j-1} 2^{-l_k} < \sum_{k=1}^{n} 2^{-l_k} \le 1$.

Az l_j hosszúságú kódszavunk legyen w_j 2-es számrendszerbeli alakja. $w_j < 2^{l_j}$ miatt az így kapott kódszó hosszúsága maximum l_j. Amennyiben a kódszó l_j-nél rövidebb, az elejére írt 0-kal a kívánt hosszúságúra egészíthető ki. Megmutatjuk, hogy ezzel az eljárással prefix kódhoz jutottunk. Indirekt bizonyítást alkalmazunk:

Tegyük fel, hogy a kapott kód mégsem prefix. Ekkor léteznek olyan w_i és w_j számok (i<j), hogy a w_j felhasználásával kapott kódszó eleje megegyezik a w_i felhasználásával kapott kódszóval. Két esetet vizsgálunk:

Ha $i \neq 1$, akkor w_i és w_j kettes számrendszerbeli alakja is 1-essel kezdődik, így a kód prefix volta csak úgy sérülhet, ha a két szám elé ugyanannyi 0-át írtunk, amikor a megfelelő hosszúságúra egészítettük ki őket. Ekkor w_j kettes számrendszerbeli felírásának eleje megegyezik w_i kettes számrendszerbeli alakjával. Vizsgáljuk meg egy példán, mit jelent ez:

Legyen pl. w_j=100110101 és w_i=10011. A két szám hosszának különbsége 4. w_j –t $2^4 = 16$ -tal elosztva 10011,0101-et kapunk (itt "," a "kettedesvessző"). Ennek egészrésze 10011, ami éppen w_i-vel egyenlő.

Könnyen látható, hogy a fentihez hasonló összefüggés általánosan is igaz. Képlettel megfogalmazva:

$$w_i = \left\lfloor \frac{w_j}{2^{l_j - l_i}} \right\rfloor = \left\lfloor \frac{\sum_{k=1}^{j-1} 2^{l_j - l_k}}{2^{l_j - l_i}} \right\rfloor = \left\lfloor \sum_{k=1}^{j-1} 2^{l_i - l_k} \right\rfloor$$

(Itt felhasználtuk, hogy w_j és w_i elé ugyanannyi 0-t írtunk, így l_j-l_i megegyezik a két szám hosszának különbségével.)

Másrészt a definíció alapján:

$$w_i = \sum_{k=1}^{i-1} 2^{l_i - l_k}.$$

Mivel j>i, a $\sum_{k=1}^{j-1} 2^{l_i - l_k}$ összegben szerepel $2^{l_i - l_i} = 2^0 = 1$, ez a tag a $\sum_{k=1}^{i-1} 2^{l_i - l_k}$ összegben nem jelenik meg. Másrészt a $\sum_{k=1}^{i-1} 2^{l_i - l_k}$ összeg minden tagja szerepel a $\sum_{k=1}^{j-1} 2^{l_i - l_k}$ összegben, így $\sum_{k=1}^{i-1} 2^{l_i - l_k} \le \sum_{k=1}^{j-1} 2^{l_i - l_k}$ -1, amiből $w_i = \sum_{k=1}^{i-1} 2^{l_i - l_k} < \left\lfloor \sum_{k=1}^{j-1} 2^{l_i - l_k} \right\rfloor = w_i$ következik, tehát ellentmondásra jutottunk.

Ha i=1, akkor w_i felhasználásával egy csupa 0-ból álló kódszót kapunk. Így w_j l_1 db 0-val kezdődik. Azonban a w_j-t megadó összegben szerepel $2^{l_j - l_1}$, ennek kettes számrendszerbeli alakja már önmagában $l_j - l_1 + 1$ hosszúságú, ha ez elé még l_1 db 0-t írnánk, akkor l_j-nél hosszabb kódszóhoz jutnánk, ami ismét ellentmondás.

Azzal a feltételezéssel, hogy a kapott kód nem prefix, mindkét esetben ellentmondásra jutottunk, ami a lemma helyességét bizonyítja.

Az előző bizonyításban használt konstrukciót jobban megvilágíthatjuk egy példával. A 4. példában láttuk, hogy $\sum_{i=1}^{4} 2^{-l_i} \le 1$ teljesül az l_1=1,l_2=2,l_3=3,l_4=3 kódszóhosszakra. Ekkor: $w_1 = 0$, $w_2 = 2^{l_2 - l_1} = 2$,

$w_3 = \sum_{k=1}^{2} 2^{l_3 - l_k} = 2^2 + 2^1 = 6$, és $w_4 = \sum_{k=1}^{3} 2^{l_4 - l_k} = 2^2 + 2^1 + 2^0 = 7$. E négy szám kettes számrendszerbeli alakjai rendre 0, 10, 110, 111. Látható, hogy prefix kódhoz jutottunk, megfelelő kódszóhosszakkal. (Éppen visszakaptuk a 2. példában mutatott kódszavakat.) Most nem volt szükség arra, hogy a számok kettes számrendszerbeli alakjának elejére 0 számjegyeket írjunk.

A lemma bizonyítása után hozzáláthatunk eredeti tételünk bizonyításához.

A tétel bizonyítása:

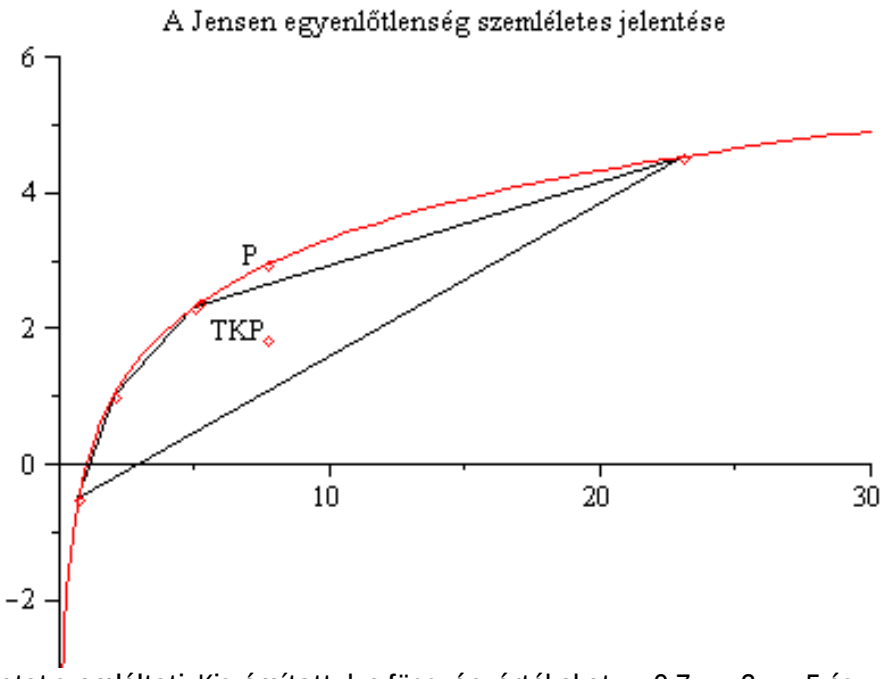
Először a $H(P) \geq \min\left(\sum_{i=1}^{n} p_i \cdot l_i\right)$ összefüggést bizonyítjuk. Vizsgáljuk a következő különbséget:

$$H(P) - \sum_{i=1}^{n} p_i \cdot l_i = \sum_{i=1}^{n} p_i \cdot \log_2 \frac{1}{p_i} + \sum_{i=1}^{n} p_i \cdot \log_2 \frac{1}{2^{l_i}} = \sum_{i=1}^{n} p_i \cdot \log_2 \frac{1}{p_i \cdot 2^{l_i}}$$

(Felhasználtuk, hogy $-l_i = \log_2 2^{-l_i} = \log_2 \frac{1}{2^{l_i}}$, és hogy az azonos alapú logaritmusok összege a szorzat logaritmusával egyezik meg.) Alkalmazzuk a Jensen egyenlőtlenséget $\sum_{i=1}^{n} p_i \cdot \log_2 \frac{1}{p_i \cdot 2^{l_i}}$-re. Eszerint a logaritmushoz hasonló konkáv függvény esetén:

$$\frac{\sum_{i=1}^{n} f(x_i)}{n} \leq f\left(\frac{\sum_{i=1}^{n} x_i}{n}\right)$$

Az egyenlőtlenség egzakt bizonyítása helyett megmutatjuk annak szemléletes jelentését:



Az ábra az n=4 esetet szemlélteti. Kiszámítottuk a függvényértékeket x_1=0,7, x_2=2, x_3=5 és x_4=23 helyeken. Az így kapott négyszög súlypontjának koordinátái:

$$\left(\frac{\sum_{i=1}^{4} x_i}{4}, \frac{\sum_{i=1}^{4} f(x_i)}{4}\right),$$

ez az ábra TKP pontja. Látható, hogy konkáv függvény esetén ez a pont a $\frac{\sum_{i=1}^{4} x_i}{4}$-hez tartozó függvényérték alá esik (ez utóbbi az ábra *P* pontja).

Alkalmazzuk a Jensen egyenlőtlenséget a logaritmusfüggvényre:

$$\sum_{i=1}^{n} p_i \cdot \log_2 \frac{1}{p_i \cdot 2^{l_i}} \le \log_2 \sum_{i=1}^{n} p_i \cdot \frac{1}{p_i \cdot 2^{l_i}} = \log_2 \sum_{i=1}^{n} 2^{-l_i} \le \log_2 1 = 0$$

Itt az utolsó egyenlőtlenségnél a Kraft-McMillan egyenlőtlenséget használtuk. Végeredményben

$H(P) - \sum_{i=1}^{n} p_i \cdot l_i \le 0$ összefüggéshez jutottunk,ami a tétel első felét bizonyítja.

Be fogjuk látni, hogy $l_i = \left\lceil \log_2 \frac{1}{p_i} \right\rceil$ kódszóhosszak esetén $\sum_{i=1}^{n} p_i \cdot l_i \le H(P)+1$ teljesül.[1] A lemma szerint ilyen kódszóhosszakkal létezik prefix (tehát egyértelműen dekódolható) kód, mivel:

$$\sum_{i=1}^{n} 2^{-\left\lceil \log_2 \frac{1}{p_i} \right\rceil} = \sum_{i=1}^{n} \frac{1}{2^{\left\lceil \log_2 \frac{1}{p_i} \right\rceil}} \le \sum_{i=1}^{n} \frac{1}{2^{\log_2 \frac{1}{p_i}}} = \sum_{i=1}^{n} p_i = 1$$

Viszont:

$$\sum_{i=1}^{n} p_i \cdot \left\lceil \log_2 \frac{1}{p_i} \right\rceil < \sum_{i=1}^{n} p_i \cdot \left(\log_2 \frac{1}{p_i} + 1 \right) = \sum_{i=1}^{n} p_i \cdot \log_2 \frac{1}{p_i} + \sum_{i=1}^{n} p_i = H(P)+1.$$

Ezzel a tételben szereplő második egyenlőtlenséget is beláttuk.

A tétel második egyenlőtlenségének bizonyításában használt $l_i = \left\lceil \log_2 \frac{1}{p_i} \right\rceil$ kódszóhosszakkal nem mindig kapunk optimális eredményt. Ez látható a Shannon tételét szemléltető 2. példából, ahol ezzel a konstrukcióval 4,3,2,2 kódszóhosszakat kapunk, pedig 3,3,2,1 hosszúságú kódszavakkal is megadható megfelelő kód.

A fenti hosszabb bizonyítás után foglalkozzunk egy kicsit azzal az úttal, melyen át üzenetünk eljut a címzetthez. Ezt a következő ábrával szemléltethetjük:

A kódoló üzenetünket a megadott kódszavak alapján kódszóvá alakítja, a dekódoló feladata felismerni, mely 0-1 sorozatot küldtük be a csatornába. Eddig csak azzal az ideális esettel foglalkoztunk, amikor üzenetünk hiba nélkül elérte a dekódolót. Ez általában nem teljesül, a fenti ábrát kiegészíthetnénk a csatornába belépő zajjal, mely üzenetünk egyes bitjeit módosíthatja. A dekódoló azért lehet képes felismerni az esetleges hibát, mert nem kerülhet be tetszőleges 0-1 sorozat a csatornába, csak azok, melyeket kódszóként kiválasztottunk. Pl. ha két kódszavunk 00 és 11, akkor a dekóder nagy valószínűséggel észreveszi, ha a csatornán való átjutás közben hiba történt (csak akkor nem, ha a 2 egymást követő bit mindegyike ellentétesre módosult). Ügyes kódolás esetén nem csak a hiba felismerése, de a hibajavítás is lehetővé válik.

Két célt kell tehát szem előtt tartanunk: alapvető elvárás, hogy üzenetünk a csatorna másik végénél is érthető legyen, az sem kívánatos azonban, hogy a küldött bitsorozat hossza túlságosan megnövekedjen. Pl. ha a rádióban a bemondó egy fontos telefonszámot csak egyszer mond el, akkor sok hallgató fogja félreérteni,

[1] A Kraft-McMillan egyenlőtlenség igazolásánál használtuk az $\lfloor x \rfloor$ jelölést, ami x alsó egészrészét, azaz a legnagyobb x-nél nem nagyobb egész számot jelentette. Most $\lceil x \rceil$ az x szám felső egészrészét, másként a legkisebb x-nél nem kisebb egész számot jelenti.

ha azonban még egyszer elismétli, akkor a hiba valószínűsége jóval kisebb lesz. Teljesen felesleges lenne viszont, ha a rádióban minden egyes mondat kétszer hangzana el, ekkor az üzenet hossza növekedne meg túlságosan. Persze a fontos információ elismétlésével még mindig nem zártuk ki a hibázás lehetőségét, a félreértés valószínűségét a lényeges adat újabb és újabb elismétlésével tovább csökkenthetjük, ekkor azonban ismét a kommunikáció sebessége csökkenne le túlságosan.

Vizsgáljunk egy *t* hosszú 0-1 sorozatot. Ha ilyen hosszúság mellett *M* db lehetséges üzenetet küldhetünk el, akkor a kommunikáció sebességét a következőképpen definiáljuk:

$$R = \frac{\log_2 M}{t}$$

Ez nyilván $M = 2^t$ esetén lenne egységnyi, ennyi különböző *t* hosszúságú üzenetet küldhetnénk, ha nem kellene a zaj miatt fellépő hibákkal foglalkozni. Ha a hibajavításra plusz biteket kell rászánnunk, akkor *R* csökken. Kódunk akkor jó, ha képesek vagyunk kis hibavalószínűség mellett is gyorsan kommunikálni. Kérdés, egy adott csatorna esetén mi az *R* sebesség szuprémuma, ha a hibázás esélyét egy előre megadott ε alá szeretnénk szorítani. (Itt a hibavalószínűséget kétféleképpen is értelmezhetjük. Egyrészt kiszámíthatjuk minden egyes kódszóra annak a valószínűségét, hogy a csatornán való átjutás után már nem (vagy rosszul) ismeri fel a dekódoló, és vehetjük ezen valószínűségek átlagát. Ezután ezt az átlagot akarjuk minimalizálni. Viszont azt is megtehetjük, hogy a különböző kódszavakra számolt hibavalószínűségek közül kiválasztjuk a maximálisat, és ennek nagyságát szeretnénk korlátozni.)

Szemléletes állítás, és hajlamosak lennénk bizonyítás nélkül elfogadni, hogy nem lehet tetszőlegesen kis hibavalószínűséggel és egyben pozitív sebességgel kommunikálni. (Fenti, rádiós bemondóval kapcsolatos példánk is azt sugallhatja, hogy az üzenet hossza a végtelenbe tart, ahogy a hiba valószínűsége 0-hoz közelít. Eközben az elküldhető üzenetek *M* száma állandó marad, a bemondó még mindig ugyannyi lehetséges telefonszám közül mond el egyet, mint korábban, *R* tehát 0-hoz tart.) Sokáig általános vélekedés volt, hogy valóban nem lehet pozitív sebességgel kommunikálni, ha a hibázás esélye 0-hoz tart. Shannon fontos felfedezése, hogy ez nem igaz: tetszőleges csatornához létezik egy küszöbszám (és ez sok fontos csatorna esetében pozitív), melynél kisebb sebességek esetén a hiba valószínűsége bármilyen kis érték alá szorítható (a küszöbszámnál nagyobb sebességek esetén pedig a hibázás valószínűsége 1-hez tart.) Ezt az éles küszöbszámot a csatorna kapacitásának nevezzük. Shannon csatornakapacitási tétele, mely az előbbi állítást kimondja, az információelmélet egyik alaptétele.

A tétel bizonyítása túlmutat ezen írás keretein. A csatornakapacitás viszont szoros kapcsolatban van egy *P* valószínűségi eloszlás *H(P)* entrópiájával, mellyel a fentiekben bővebben foglalkoztunk, ezért erről még szólunk pár szót.

Egy csatorna (legalábbis egy ún. diszkrét emlékezetnélküli csatorna, mi most csak ilyenekkel foglalkozunk) megadása a következő táblázattal lehetséges:

A táblázat minden sora elé odaírjuk a bemeneti ABC (azaz a bemeneti jelkészlet) egy-egy betűjét, oszlopainak pedig hasonlóan a kimeneti ABC egy-egy betűjét feleltetjük meg. (Az eddigiekben csak bináris kódokkal foglalkoztunk, ilyenkor az ABC a 0 és 1 jelekből áll.) Az i-edik sor j-edik oszlopába az a p valószínűség kerül, mely megmondja, az i-edik sorhoz tartozó bemenet esetén mekkora valószínűséggel kapjuk a j-edik oszlophoz írt kimenetet. Tekintsük a következő egyszerű példát:

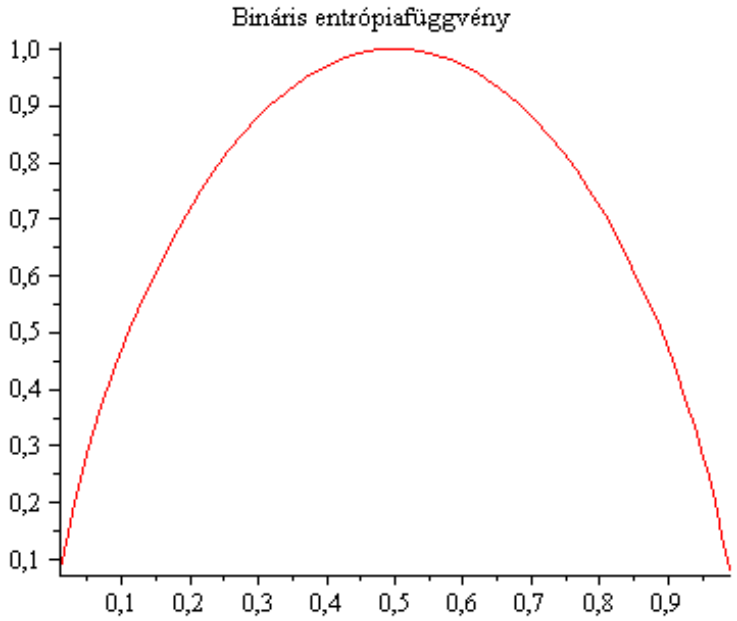
	0	1
0	1-p	p
1	p	1-p

A be- és a kimeneten egyaránt a 0 vagy az 1 bit jelenhet meg. Ha a bemeneten 0-t viszünk be, akkor a kimeneten 1-p valószínűséggel 0-t, p valószínűséggel 1-et kapunk vissza. Hasonlóan a bemeneti 1-es a kimeneten p valószínűséggel 0-t, 1-p valószínűséggel 1-est eredményez. Ideális esetben p=0, ekkor a csatornán át pluszbitek használata nélkül hibamentesen tudunk kommunikálni. Ha p elég kicsi, a hiba valószínűsége kicsi lesz. Ugyanilyen kedvező számunkra, ha p közel van 1-hez (ilyenkor a bemeneti bit nagy valószínűséggel a kimeneti bit ellentettje volt). p=0,5 esetén viszont a csatorna teljesen használhatatlan, a kimeneten kapott bit tökéletesen független a bemenettől.

Az adott csatornán át nyilván akkor tudunk jól kommunikálni, ha van néhány olyan bemenet, ami nagy valószínűséggel olyan kimenetet eredményez, ami jó közelítéssel csak az adott bemeneti jelből keletkezhet.

A csatorna be- és kimenetéhez is tartozik egy X, illetve Y, be-, illetve kimeneti ABC. A bemeneti jeleket értelmezhetjük egy értékeit X -en felvevő valószínűségi változóként.[2] (Persze X eloszlását mi határozzuk meg a kódszavak megválasztásával. Az egyes üzenetek küldésének valószínűsége a hozzájuk tartozó események bekövetkezésének valószínűségével egyezik meg, viszont mi rendeljük hozzá a kódszavakat az egyes üzenetekhez.) Ekkor a kimeneten is kapunk egy Y -on adódó valószínűségeloszlást. Kiszámíthatjuk az X valószínűségi változó eloszlásának fentebb definiált, most egyszerűen H(X)-szel jelölt entrópiáját. Ezután valamely "a" bemeneti jel esetén is kiszámolhatjuk Y entrópiáját, ezt a feltételes entrópiát H(Y|X=a)-val jelöljük. Ha a számítást az összes "a" bemenetre elvégezzük, és a kapott eredményeket átlagoljuk, akkor a H(Y|X) entrópiához jutunk, ami szemléletesen kifejezve annak a mértéke, mennyire marad bizonytalan Y, ha X-et már ismerjük. Másképpen fogalmazva: H(Y|X) megmutatja, mennyi információt hordoz Y megfigyelése még azután is, hogy X-et már tudjuk. Az Y valószínűségi változó eloszlásának is van egy H(Y) entrópiája, a H(Y)-H(Y|X) különbség tehát megadja, mennyit árul el X Y-ról. Ha ezt elfogadjuk, akkor szemléletes az az állítás, hogy nyilván ugyannyit, mint amennyit Y árul el X-ről. Tehát H(Y)-H(Y|X)=H(X)-H(X|Y). Ezt a mennyiséget az X és Y valószínűségi változók *kölcsönös információ*jának nevezzük, és I(X,Y)-nal jelöljük. A Shannon féle csatornakapacitási tételben szereplő küszöbszám (azaz a csatorna kapacitása) éppen $C=\max_{P_X} I(X,Y)$. (Nyilván az a célunk, hogy Y minél többet áruljon el X-ről. A bemenethez tartozó X eloszlást mi választhatjuk meg, ezt kell úgy variálnunk, hogy I(X,Y) maximális legyen.)

Érdemes még megemlíteni, hogy a fenti egyszerű 2x2-es táblázattal megadott csatorna esetén I(X,Y) maximális értéke az $\frac{1}{2}$- $\frac{1}{2}$ bemeneti eloszláshoz tartozik. Nyilván az a=1 és a=0 bemenet esetén is H(Y|X=a)=H(p,1-p), így az átlagolás után kapjuk: H(Y|X)=H(p,1-p). H(p,1-p)-t h(p)-vel is szokás jelölni, ennek értéke csak p-től függ. Célunk H(Y)-h(p) maximalizálása. H(Y) egy két értéket felvevő valószínűségi változó entrópiafüggvénye, az úgynevezett bináris entrópiafüggvény,ami tehát $q\times\log_2\frac{1}{q}+(1-q)\times\log_2\frac{1}{1-q}$ alakban írható, ahol q a két kimeneti jel egyike megjelenésének valószínűsége a kimeneten. A h(q) bináris entrópiafüggvény q=$\frac{1}{2}$ -nél veszi fel maximumát, ennek értéke 1, ami az ábráról is leolvasható:



Tehát I(X,Y) maximális értéke 1-h(p).

Ajánló

[2] A bemeneti ABC-t X-szel, az itt említett valószínűségi változót X-szel fogjuk jelölni. Ugyanígy különböző az Y, és az alább szintén használt Y jelölés jelentése.

· Rényi Alfréd: *Ars Mathematica* (benne: *Az információ matematikai fogalmáról*), Typotex, 2005
http://www.typotex.hu/konyv/Ars%20Mathematica
· Warren Weaver, Claude Shannon: *A kommunikáció matematikai elmélete*. Az információ elmélet születése és távlatai, Budapest, 1986, Országos Műszaki Információs Központ és Könyvtár
· Györfi László, Győri Sándor, Vajda István: *Információ és kódelmélet*
http://books.google.hu/books?id=0gfLaDnXmXEC
· Benczúr András: *Számítógépek és híradástechnika: az emberiség új kommunikációs korszaka*
http://davidalb.web.elte.hu/infkez4/Benczurjegyzet1.doc
http://davidalb.web.elte.hu/infkez4/Benczurjegyzet2.doc
· Petz Dénes: Neumann János és a kvantumbitek, előadásjegyzet
www.renyi.hu/~petz/pdf/Fazekas.pdf

A klasszikus információelmélettel való ismerkedés után térjünk át egy érdekes kapcsolat ismertetésére az információelmélet és a gráfelmélet között. Gyakorlati szempontból általában teljesen kielégítő, ha tetszőlegesen kicsi hibával tudunk kommunikálni, mégis feltehetjük a kérdést: lehetséges-e a hibátlan kommunikáció?

Térjünk vissza a csatornánkat megadó táblázathoz. A fenti egyszerű példánál $p = 0{,}1$ esetén a 0 hibával való kommunikáció nyilván teljesen reménytelen. Lássunk egy másik példát:

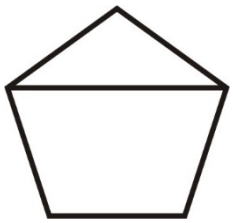
	y_1	y_2	y_3	y_4	y_5
x_1	0,1	0,2	0,4	0,2	0,1
x_2	0	0	0,7	0	0,3
x_3	0,5	0,1	0,1	0,2	0,1
x_4	0,1	0,6	0	0,3	0
x_5	0,2	0,5	0,1	0,1	0,1

Nyilvánvaló, hogy ilyen csatorna esetén az x_2 és az x_4 bemeneti jelet mindig meg tudja különböztetni egymástól a dekódoló, hiszen x_4-ből nem keletkezhet sem az y_3, sem az y_5 kimenet, x_2 bemenet esetén pedig a kimeneten biztosan ezek egyike jelenik meg. Kicsit általánosabban fogalmazva: ha két bemeneti jel olyan, hogy a táblázat bármely oszlopát vizsgálva maximum egyiküknek a sorában áll pozitív érték, akkor ezt a két jelet biztosan meg tudjuk különböztetni egymástól. Ha viszont nem létezik legalább két olyan bemeneti jel, melyek garantáltan nem téveszthetők össze, akkor biztosan nem lehetséges 0 hibával kommunikálni csatornánkon keresztül.

Azt, hogy két jelet meg lehet-e egymástól különböztetni, könnyen leírhatjuk a következő gráffal: a gráf csúcsai legyenek a bemeneti jelek, két csúcs között akkor fusson él, ha a csúcsokhoz rendelt jelek biztosan nem téveszthetők össze. Az így kapott gráfot a csatorna bemeneti betűihez tartozó megkülönböztethetőségi gráfnak nevezzük (az összetéveszthetőségi gráf, mely ennek komplementere, teljesen hasonlóan definiálható). Kérdés, egy adott G megkülönböztethetőségi gráfú csatorna esetén hány olyan t hosszúságú kódszót készíthetünk, melyek páronként megkülönböztethetők. Ha M a megfelelő kódszavak lehetséges maximális száma, akkor ismét a $\frac{\log_2 M}{t}$ hányados határértékét keressük $t \to \infty$ esetén. Ezt a számot a csatorna zéróhiba-kapacitásának nevezzük.

Ennek vizsgálatához érdemes elkészíteni a G gráf megfelelőjét a bemeneti jelekből alkotott t hosszú sorozatokra, ezt G^t-vel jelöljük. G^t csúcsai az egyes t hosszú sorozatok, két csúcs között akkor fut él, ha a hozzájuk rendelt sorozatok egymástól megkülönböztethetők. Ez akkor következik be, ha a két sorozat legalább egyetlen helyen nem összetéveszthető, azaz legalább egy helyen a G gráf élét alkotta.

Egy F gráf fontos jellemzője az $\omega(F)$ klikkszám, amely F legnagyobb teljes részgráfjának mérete. (Teljes gráfról akkor beszélünk, ha minden csúcs az összes többi csúccsal össze van kötve.) Nyilván maximum annyi páronként megkülönböztethető t hosszúságú sorozatot készíthetünk, amennyi G^t klikkszáma. Pl. tekintsük a következő megkülönböztethetőségi gráfhoz tartozó csatornát:



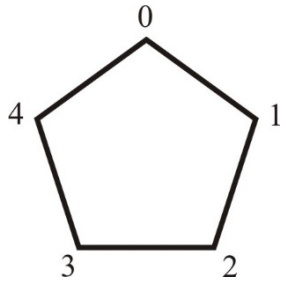
Itt G legnagyobb teljes részgráfja egy háromszög, $\omega(G)=3$, így maximum 3 páronként megkülönböztethető 1 hosszú üzenetet küldhetünk csatornánkon át. Ahhoz, hogy a páronként megkülönböztethető t hosszú sorozatok számát meghatározzuk, az $\omega(G^t)$ klikkszámot kellene ismernünk.

Eredeti problémánkat így tisztán gráfelméleti feladattá fogalmazhatjuk át: a G gráf Shannon kapacitását akarjuk meghatározni, melyet a következőképpen definiálunk[3]:

$$C(G)=\lim_{t\to\infty}\frac{\log_2 \omega(G^t)}{t}.$$

A Shannon kapacitás olyan gráfparaméter, melynek kiszámítása komoly nehézségekbe ütközik. Nem egyszerűen arról van szó, hogy egy nagy méretű gráf esetén nehéz algoritmust találni a meghatározására (sok gráfparaméter esetén ez a helyzet), az sem ismert, hogy egyáltalán létezik-e megfelelő algoritmus. Már egészen kis csúcsszámú gráfoknál előfordulnak olyan esetek, amikor a gráf Shannon kapacitását nem ismerjük.

A G gráf Shannon kapacitásának nyilvánvaló alsó korlátja $\log_2 \omega(G)$, ugyanis ha üzeneteinket csak azokból a bemeneti jelekből állítjuk elő, melyek páronként megkülönböztethetőek, akkor nyilván páronként össze nem téveszthető t hosszú sorozatokat kapunk. Ilyen sorozatokból éppen $(\omega(G))^t$ készíthető, így valóban $C(G)=\lim_{t\to\infty}\frac{\log_2 \omega(G^t)}{t} \geq \lim_{t\to\infty}\frac{\log_2(\omega(G))^t}{t}=\log_2 \omega(G)$. Kérdés, fölé lehet-e menni ennek az alsó korlátnak. A válasz igen, a legkisebb csúcsszámú olyan gráf, melyben a fenti kifejezésben szigorú egyenlőtlenség áll, az 5 hosszúságú kör:



Ennek a gráfnak a klikkszáma 2, így ha csak páronként megkülönböztethető bemeneti jeleket használunk a 2 hosszú sorozatokban, akkor csak $2^2=4$ sorozatot készíthetünk. Ügyesebb módszerrel 5 páronként össze nem téveszthető 2 hosszúságú sorozatot is megadhatunk:

00, 12, 24, 31, 43.

A felsorolásban egymás után következő üzenetek (valamint az utolsó és az első üzenet) első betűi össze nem téveszthetők, a felsorolásban másodszomszédos sorozatoknak (illetve a negyedik és az első, valamint az utolsó és a második sorozatnak) pedig a második betűje megkülönböztethető, így az összes sorozat páronként megkülönböztethető. Ha $t=2\times k$, akkor ezt az öt 2 hosszúságú sorozatot felhasználva 5^k darab t hosszú sorozatot készíthetünk, melyek páronként megkülönböztethetőek. Ebből azonnal adódik, hogy az 5 hosszú kör kapacitása legalább $\sqrt{5}$. Sokáig nyitott kérdés maradt, ez-e az optimális érték. Lovász László egyik híres eredménye annak a bizonyítása, hogy az 5 hosszúságú kör Shannon kapacitása valóban $\sqrt{5}$-tel egyenlő. (Ennek jelentősége abban is áll, hogy a bizonyítás során Lovász bevezetett egy később nagyon fontossá vált gráfparamétert.)

[3] Megjegyezzük, hogy sok tárgyalás az összetéveszthetőségi gráf segítségével, ezért komplementer módon definiálja ezt a fogalmat, vagyis úgy, hogy amit mi C(G)-vel jelölünk, azt ezen tárgyalások a komplementer gráf Shannon kapacitásának nevezik.

Ajánló

· Lovász László: On the Shannon capacity of a graph
http://www.cs.elte.hu/~lovasz/scans/theta.pdf

· Tom Bohman, Ron Holzman: A nontrivial lower bound on the Shannon capacities of the complements of odd cycles
http://www2.technion.ac.il/~holzman/papers/completr.pdf

· Noga Alon: The Shannon capacity of a union
http://www.math.tau.ac.il/~nogaa/PDFS/shann3.pdf

Megmutattuk, hogy egy G gráf Shannon kapacitása alulról becsülhető $\log_2 w(G)$-vel. Könnyen bebizonyítható, hogy C(G)-re felső korlát lesz $\log_2 c(G)$, a gráf kromatikus számának logaritmusa.

Egy gráf kromatikus száma azt adja meg, legalább hány színre van szükség a gráf csúcsainak olyan színezéséhez, ahol az egymással szomszédos csúcsok (azok a csúcsok, melyek között fut él) különböző színűek. Nyilvánvaló, hogy $w(G) \le c(G)$ mindig teljesül, hiszen a gráfban van $w(G)$ darab páronként összekötött csúcs, melyek közül semelyik kettő nem lehet azonos színű. Mielőtt bebizonyítanánk, hogy $C(G) \le \log_2 c(G)$, lássuk be a következő segédtételt:

Lemma:
$$c(G^t) \le (c(G))^t$$

A lemma bizonyítása:
Vegyük G egy jó színezését $c(G)$ színnel. Ennek segítségével fogjuk definiálni a G^t gráf egy megfelelő színezését $(c(G))^t$ színnel. G^t minden csúcsa egy-egy G csúcsaiból álló sorozat. Egy ilyen csúcsot színezzünk a G-beli megfelelő színek sorozatával. Ha két G^t –beli csúcs össze van kötve, akkor valamely i-re a csúcsok i-edik koordinátájában G-ben összekötött csúcsok állnak, és így ezek színe különböző az eredeti gráf színezésében. Tehát itt a színsorozat is eltér, vagyis G^t –ben az összekötött csúcsokat különböző színekkel színeztük. A $c(G)$ színből legfeljebb $(c(G))^t$ különböző t hosszúságú színsorozat készíthető, tehát legfeljebb ennyi színt használtunk fel G^t jó színezéséhez. Így G^t legfeljebb $(c(G))^t$ színnel kiszínezhető, kromatikus száma nem lehet ennél nagyobb. Ezzel a lemma állítását beláttuk.

Ezután lássuk a tétel bizonyítását:

Tétel:
$$C(G) \le \log_2 c(G)$$

Bizonyítás:
$$w(G^t) \le c(G^t) \le (c(G))^t$$
Vonjunk t-edik gyököt az egyenlőtlenség két oldalából, és vegyük a kettes alapú logaritmust:
$$\log_2(w(G^t)) \le t \cdot \log_2 c(G)$$
Osszunk t-vel, és vizsgáljuk mindkét oldal határértékét $t \to \infty$ esetén. A bal oldalon definíció szerint C(G) fog állni, így valóban:
$$C(G) \le \log_2 c(G).$$

Végeredményben a következő egyenlőtlenséglánchoz jutottunk:
$$\log_2 w(G) \le C(G) \le \log_2 c(G).$$

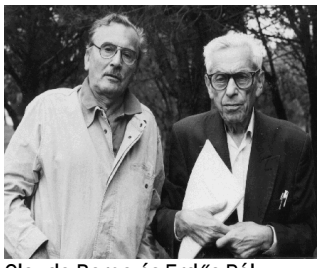
Ha $w(G) = c(G)$, ez lehetőséget nyújt arra, hogy C(G)-t meghatározzuk, de sok gráf esetén ez az egyenlőség nem áll. A legegyszerűbb olyan gráf, ahol nem teljesül az egyenlőség, éppen a fenti példánál látott 5 hosszúságú kör.

Vajon milyenek lehetnek azok a gráfok, melyekre $w(G) = c(G)$ teljesül? Ezzel a kérdéssel az 1960-as évek elején kezdett el foglalkozni Claude Berge francia matematikus, éppen a Shannon-féle gráfkapacitás fenti tulajdonsága által inspirálva. Az, hogy egy G gráfnál ez a két gráfparaméter egyenlő, még nem mond sokat a gráf struktúrájáról, hiszen ha tekintünk egy olyan gráfot, ahol ez a két érték nagyon eltér egymástól, majd mellérakunk egy elég nagy teljes gráfot, akkor az így kapott G gráfban $c(G) = w(G)$ teljesülni fog. Ezért Berge a következő kikötést tette: vegyük azokat a gráfokat, amelyek esetén mind az eredeti gráfra, mind annak összes feszített részgráfjára fennáll az $c(G) = w(G)$ egyenlőség (feszített részgráfhoz akkor jutunk, ha a gráf bizonyos pontjait, valamint valamennyi ezen csúcsok között futó élt kijelöljük, a többi csúcsot és élt pedig elhagyjuk). Az ilyen tulajdonságú gráfokat Berge perfekt gráfoknak nevezte el. Az évek során kiderült, hogy ezek igen fontos, sok érdekes tulajdonsággal rendelkező gráfosztályt alkotnak.

Berge megfogalmazta azt a sejtést, hogy ha egy gráf perfekt, akkor a komplementere is az. Lovász egy másik híres eredménye ennek a sejtésnek a bizonyítása. Egy az előzőnél erősebb, szintén Berge-től származó sejtés évtizedekig nyitva maradt, több mint 150 oldalas cikkben megjelent bizonyítása csak pár éve született meg: pontosan azok a gráfok perfektek, melyek esetén sem a gráf, sem a komplementere nem tartalmaz feszített részgráfként (vagyis 'átlói' nélkül) legalább 5 hosszúságú páratlan kört. Ezen tétel bizonyítása (melyet Maria Chudnovsky, Neil Robertson, Paul Seymour és Robin Thomas talált meg) az utóbbi évek egyik legnagyobb gráfelméleti eredménye.

Láttuk, hogy az információelmélet nem csak önmagában jelentős ága a matematikának, hanem más területekre is inspirálóan hatott. A Shannon által bevezetett fogalmak az eredeti problémától igen messzire vezettek, és például a gráfelméletben is számos új, érdekes eredmény megszületését inspirálták.

Ajánló



Claude Berge és Erdős Pál
Chronomaths
http://serge.mehl.free.fr/

· A 2004. évi párizsi Claude Berge emlékülés oldala
http://www.ecp6.jussieu.fr/GT04/Berge/Berge.html
· Vašek Chvátal: Claude Berge 5. 6. 1926 – 30. 6. 2002
http://users.encs.concordia.ca/~chvatal/perfect/claude2.pdf
· Denis Boyssou, Dominique de Werra, Olivier Hudry: Claude Berge and the „Oulipo"
http://www.lamsade.dauphine.fr/~bouyssou/Berge.pdf
· Recski András: Gráfok színezése
http://matek.fazekas.hu/portal/eloadas/2007/eloadas_2008_01_22_recski.html
· Wikipédia a perfekt gráfokról:
http://en.wikipedia.org/wiki/Perfect_graph
· M. Chudnovsky, N. Robertson, P. D. Seymour, R. Thomas: Progress on perfect graphs,
http://people.math.gatech.edu/~thomas/PAP/perfsur.pdf
· Paul Seymour: How the proof of the strong perfect graph conjecture was found
http://www.math.princeton.edu/~pds/papers/howtheperfect/howtheperfect.pdf
· Tony Jebara előadása a Perfekt gráfokról:
http://videolectures.net/mlss09us_jebara_mapepg/
· Vašek Chvátal: Perfect problems
http://users.encs.concordia.ca/~chvatal/perfect/problems.html
· Simonyi Gábor: Graph Entropy – A Survey
www.renyi.hu/~simonyi/grams.ps