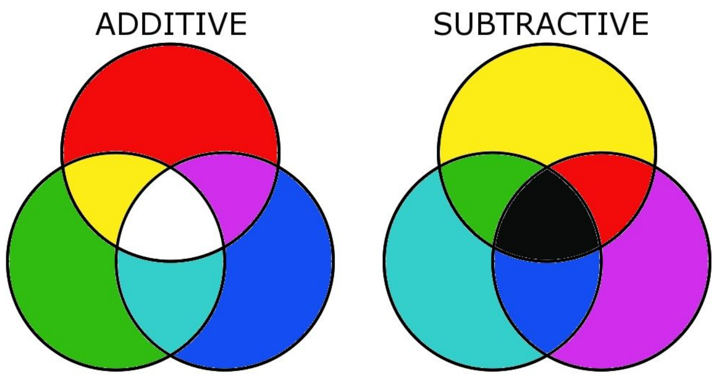
576 Fall22 HW2
Name: Sung-Fu Han
USC ID: 2644230653

Q1-1



Based on subtractive color theory, in the blue light picture, the light is composed by blue light and several white light. The blue light is everywhere and the yellow surface will absorb blue light and reflect yellow light when a small portion of white light is on it. So we see blue and yellow light at the same time, and it comes to glow white light again. The red surface reflects little red combined with blue light, so actually we see dark purple.

Q1-2
Yellow is better than red since yellow light has multiple primary colors of light.
yellow light = red + green light: 5 colors under yellow light becomes 4 colors

surface	absorb	reflect	actual see
red	blue, green	red	red
yellow	blue	red, green	yellow
blue	red, green	blue	black
green	red, blue	green	green
white	N/A	all	yellow

red light: 5 colors under yellow light becomes 3 colors

surface	absorb	reflect	actual see
red	blue, green	red	red
yellow	blue	red, green	red
blue	red, green	blue	black
green	red, blue	green	green
white	N/A	all	red

Q2-1
given
x = X/(X+Y+Z)
y = Y/(X+Y+Z)
z = Z/(X+Y+Z)

1 - x - y = 1 - X/(X+Y+Z) - Y/(X+Y+Z) = Z/(X+Y+Z) (1)
y = Y/(X+Y+Z) (2)
(1)/(2) => (1-x-y)/y = Z/Y => [(1-x-y)/y]Y = Z **#QED**

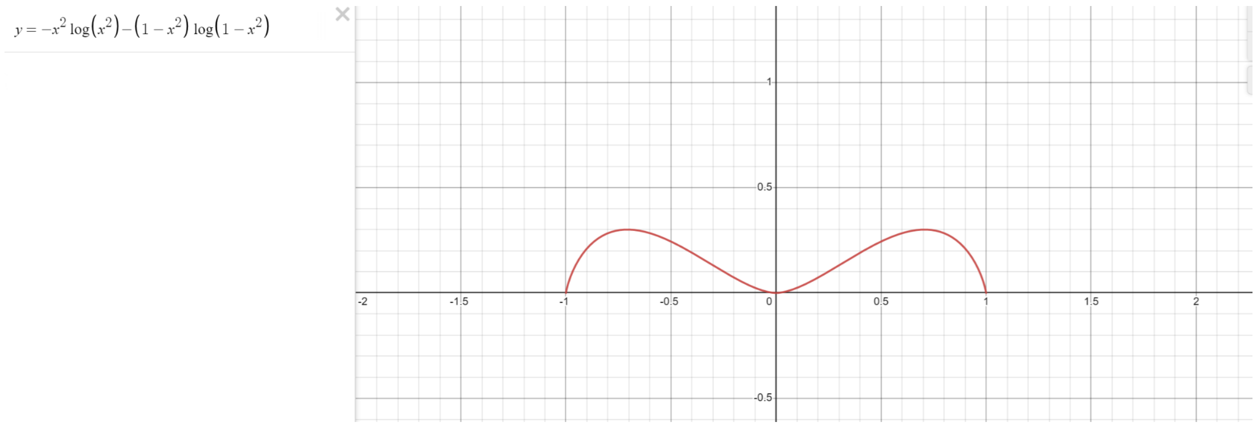
Q2-2
Mapping all out-of-gamut colors onto the nearest point of target gamut is called clipping. Unfortunately, the method is **<u>not effective</u>**, because it may **shift the wavelength from the original one** and cause color twisted, loss of detail, loss of colourfulness, loss of contrast and change of appearance.
[Ref]: https://opg.optica.org/oe/fulltext.cfm?uri=oe-26-9-11481&id=385750

Q2-3
Cartoon tones with less variation use fewer kinds of color, so there is less chance that the color falls out of the target gamut. Less transformation keeps more original colors. Therefore, **<u>cartoon tones perform better.</u>**

Q2-4
Instead of using a clipping method which may lead to blocking issues, we can choose compression methods. The most well-known compression algorithm is the <u>chroma-dependent sigmoidal lightness mapping algorithm</u> followed by <u>knee scaling toward the cusp</u>. This algorithm, together with the hue-angle preserving minimum color difference (HPMINDE) algorithm, were recommended by the International Commission on Illumination (CIE).
[Ref]: https://opg.optica.org/oe/fulltext.cfm?uri=oe-26-9-11481&id=385750

Q3-1

$H = - \mathrm{sum}_i (p_i * \log(p_i)) = - x^k * \log(x^k) - (1-x^k) * \log(1 - x^k)$
$k = 2$
$H = - x^2 * \log(x^2) - (1-x^2) * \log(1 - x^2)$



source: https://www.desmos.com/calculator?lang=zh-TW

Q3-2 Find Min
According to the picture above, **$x = 0$** H has minimal value 0.
It means the transmitter always sends the Y symbol, so entropy is 0.

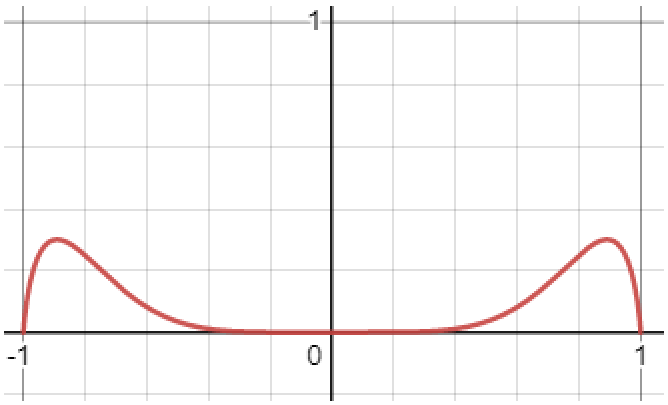
Q3-3 Find general Min
$x = 0^k$ s.t. H is 0
No matter k is, if x is 0, the transmitter always sends the Y symbol and entropy is 0.

Q3-4 Find Max
$H = - x^2 * \log(x^2) - (1-x^2) * \log(1 - x^2)$
$dH/dx = - 2x * \log(x^2) \underline{- 2x^2 * (1/x)} - (-2x) * \log(1 - x^2) \underline{- (1-x^2) * [1/(1-x^2) * -2x]}$
$= - 2x * \log(x^2) \underline{-2x} + 2x * \log(1 - x^2) \underline{+ 2x}$
$= 2x [\log(1 - x^2) - \log(x^2)]$
$= 2x * \log(1/x^2 - 1)$

$\log(1/x^2 - 1) = 0$
$\Rightarrow 1/x^2 -1 = 1$
$\Rightarrow x^2 = 1/2$
$\Rightarrow$ **$\underline{x = (1/2)^{1/2}, - (1/2)^{1/2}}$**

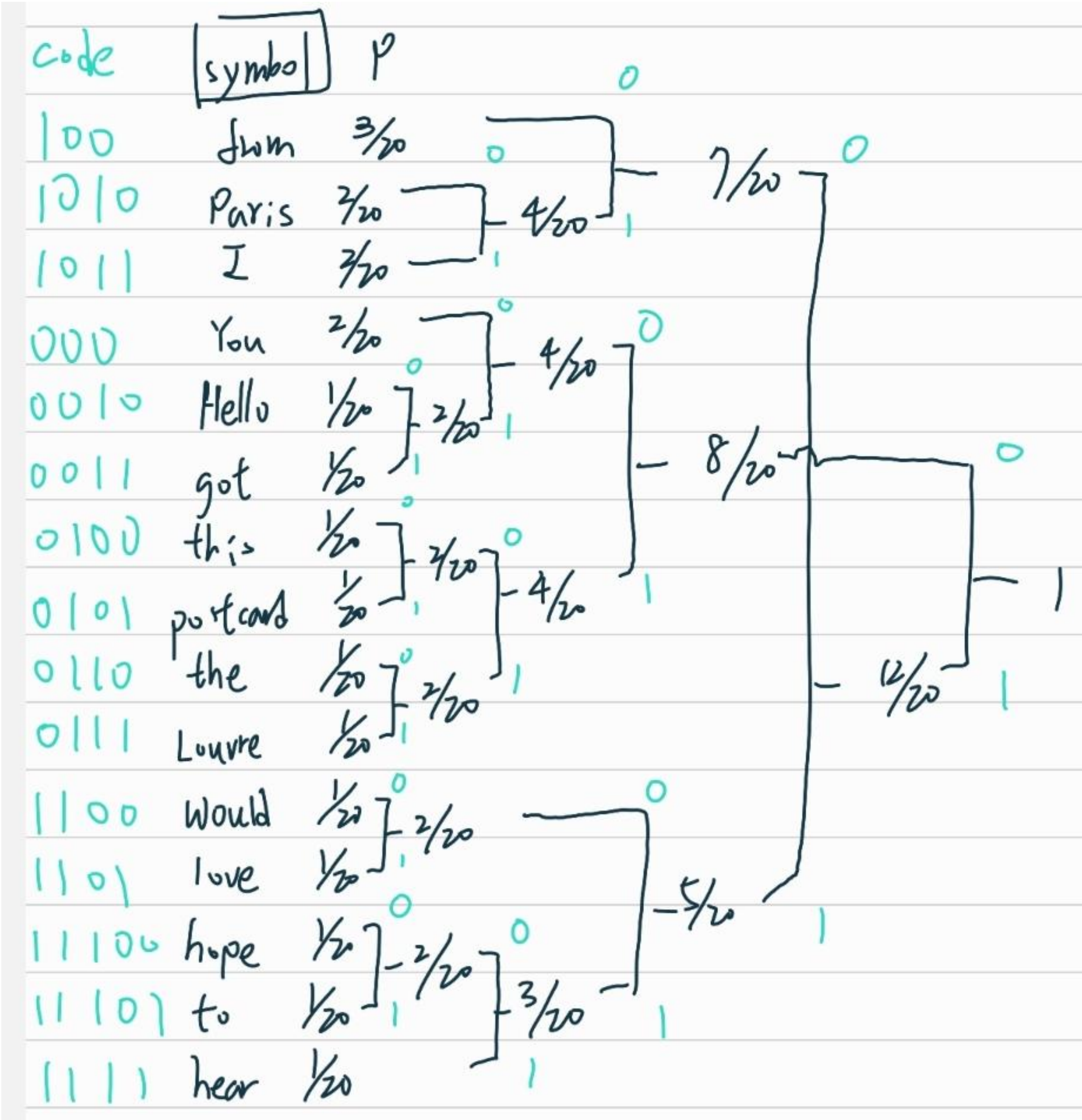
Q3-5 Find general Max
$H = - x^k * \log(x^k) - (1-x^k) * \log(1 - x^k)$
dH/dx
$= - kx^{(k-1)} * \log(x^k) \underline{- kx^k * (1/x)} - [-kx^{(k-1)}] * \log(1 - x^k) \underline{- (1-x^k) * [1/(1-x^k) * -kx^{(k-1)}]}$
$= - kx^{(k-1)} * \log(x^k) \underline{-kx^{(k-1)}} + kx^{(k-1)} * \log(1 - x^k) \underline{+ kx^{(k-1)}}$
$= kx^{(k-1)} [\log(1 - x^k) - \log(x^k)]$
$= kx^{(k-1)} * \log(1/x^k - 1)$

log(1/x^k - 1) = 0
=> 1/x^k -1 = 1
=> x^k = ½
=> x = (½)^(1/k), - (½)^(1/k) if k is even



x = (½)^(1/k) if k is odd

Q4-1

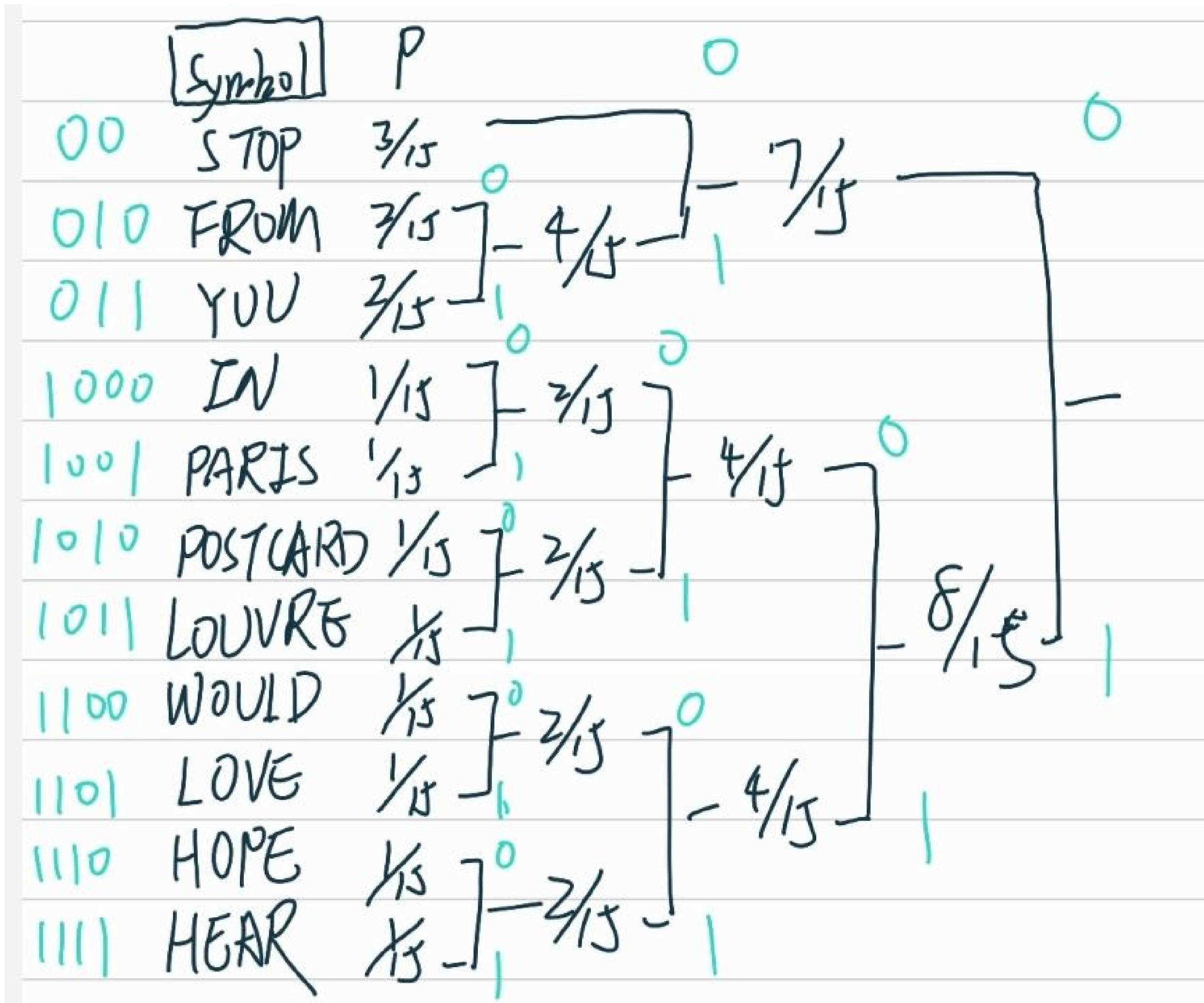


symbol	f	P_i	code	l_i
from	3	3/20	100	3
Paris	2	2/20	1010	4
I	2	2/20	1011	4
You	2	2/20	000	3

Hello	1	1/20	0010	4
got	1	1/20	0011	4
this	1	1/20	0100	4
postcard	1	1/20	0101	4
the	1	1/20	0110	4
Louvre	1	1/20	0111	4
would	1	1/20	1100	4
love	1	1/20	1101	4
hope	1	1/20	11100	5
to	1	1/20	11101	5
hear	1	1/20	1111	4

> **average code length** = sum_i(p_i * l_i) = 3/20 * 3 + 2/20 * 4 + 2/20 * 4 + …+ 1/20 * 4
= **3.85 bits #**
Entropy H = - sum_i (p_i * log(p_i)) = - 3/20 * log(3/20) - 2/20 * log(2/20) - …. - 1/20 * log(1/20)
= 3.78

Q4-2



symbol	f	P_i	code	l_i
STOP	3	3/15	00	2
FROM	2	2/15	010	3
YOU	2	2/15	011	3
IN	1	1/15	1000	4
PARIS	1	1/15	1001	4
POSTCARD	1	1/15	1010	4
LOUVRE	1	1/15	1011	4
WOULD	1	1/15	1100	4

LOVE	1	1/15	1101	4
HOPE	1	1/15	1110	4
HEAR	1	1/15	1111	4

> **average code length** = sum_i(p_i * l_i) = 3/15 * 2 + 2/15 * 3 + 2/15 * 3 + …+ 1/15 * 4
= **<u>3.33</u> bits** #
Entropy H = - sum_i (p_i * log(p_i)) = - 3/15 * log(3/15) - 2/15 * log(2/15) - …. - 1/15 * log(1/15)
= 3.32

Q4-3
In a quantitative sense, the postcard message contains more words than telegram's one.
In a qualitative sense, entropy of the postcard message is 3.78, and the entropy of the telegram message is 3.32. Therefore the **postcard message contains more information**.