HW1 Solutions ECE 563 Fall 2009

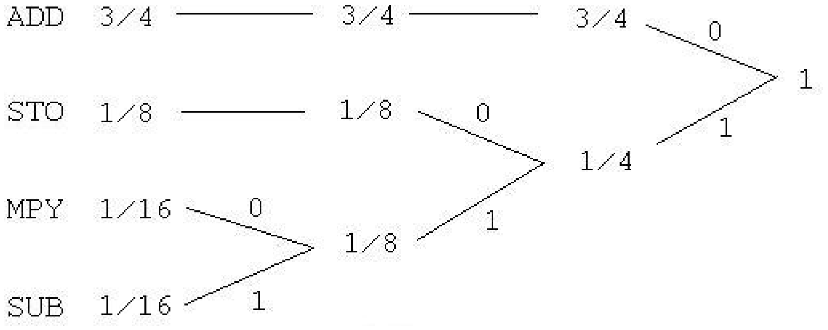
2.
$$\begin{aligned}
\frac{q'_k}{q_k} &= \left(\frac{p'}{p}\right)^k \left(\frac{1-p'}{1-p}\right)^{n-k} \\
L(q';q) &= \sum_{k=0}^{n} q'_k \log\left(\frac{q'_k}{q_k}\right) = \sum_{k=0}^{n} q'_k \left[k\log\left(\frac{p'}{p}\right) + (n-k)\log\left(\frac{1-p'}{1-p}\right)\right] \\
&= \log\left(\frac{p'}{p}\right)\sum_{k=0}^{n} kq'_k + \log\left(\frac{1-p'}{1-p}\right)\sum_{k=0}^{n}(n-k)q'_k \\
&= np'\log\left(\frac{p'}{p}\right) + n(1-p')\log\left(\frac{1-p'}{1-p}\right) \\
&= nL(p';p)
\end{aligned}$$

3. Let $f(x) = \log_e x - (x-1)$, $x\epsilon(0,\infty)$. $f'(x) = \frac{1}{x} - 1 = 0 \Rightarrow x = 1$ is a local extremum.
$f''(x)|_{x=1} < 0 \Rightarrow x = 1$ is a strict local maximum. Since $f(x)$ only has 1 extremal point in $(0,\infty)$, $x = 1$ is a strict global maximum.
$f(1) = 0 \Rightarrow f(x) \leq 0\ \forall x\ \epsilon\ (0,\infty) \Rightarrow \log_e x \leq x - 1$.

Substituting $\frac{1}{x}$ for x, we get $\log_e \frac{1}{x} \leq \frac{1}{x} - 1 \Rightarrow \log_e x \geq 1 - \frac{1}{x}$

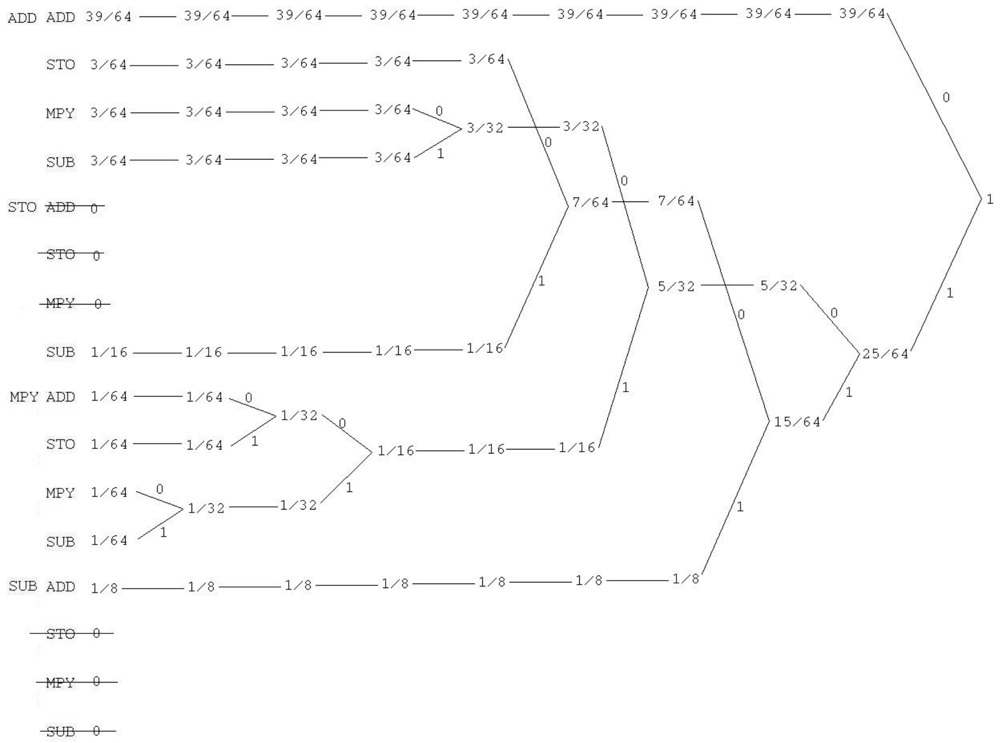
5.
$$\begin{aligned}
P(X,Y,Z) &= P(X)P(Y|X)P(Z|X,Y) = P(X)P(Y|X)P(Z|Y) = P(X,Y)P(Z|Y) \\
&= P(Y)P(X|Y)P(Z|Y) = P(Y,Z)P(X|Y) \\
P(X,Y,Z) &= P(Z)P(Y|Z)P(X|Y,Z) = P(Y,Z)P(X|Y,Z) \\
&\Rightarrow P(X|Y,Z) = P(X|Y) \Rightarrow Z \to Y \to X
\end{aligned}$$

6. a) $H(\frac{3}{4}, \frac{1}{16}, \frac{1}{16}, \frac{1}{8}) = -\left(\frac{3}{4}\log\frac{3}{4} + \frac{1}{16}\log\frac{1}{16} + \frac{1}{16}\log\frac{1}{16} + \frac{1}{8}\log\frac{1}{8}\right) = 1.186$ So the average number of bits used could be reduced by about 40.7%.
b) (solution is not unique)



So the codewords are ADD ↔ 0, SUB ↔ 110, MPY ↔ 111, and STO ↔ 10.
c)$H = -\sum_{i,j} p_j P_{i|j}\log_2 P_{i|j} = 0.87$ bits/instruction. So the average number of

bits used could be reduced by $\sim 57\%$.
d)(solution is not unique)



So the codewords are:

$$\begin{array}{rcl}
\text{ADD ADD} & \leftrightarrow & 0 \\
\text{ADD SUB} & \leftrightarrow & 1001 \\
\text{ADD MPY} & \leftrightarrow & 1000 \\
\text{ADD STO} & \leftrightarrow & 1100 \\
\text{STO SUB} & \leftrightarrow & 1101 \\
\text{MPY ADD} & \leftrightarrow & 10100 \\
\text{MPY SUB} & \leftrightarrow & 10101 \\
\text{MPY MPY} & \leftrightarrow & 10110 \\
\text{MPY STO} & \leftrightarrow & 10111 \\
\text{SUB ADD} & \leftrightarrow & 111
\end{array}$$

7. a) If we look at the entropy of a single card, it is $H_b(\frac{1}{4})$, because there is a probability of $\frac{1}{4}$ that the card will be in the hand. Although the cards are not independent, if we approximate the total entropy by assuming independence, we get $H = 52H_b(\frac{1}{4})$. A simple representation using 52 bits is to allocate one

bit to each card and let it be a 0 if the card is not in the hand or a 1 if the card is in the hand.
The actual entropy can be calculated by first determining the number of possible hands $N = \frac{52!}{39!13!}$ and then assuming they are each equally likely, so that $H = \log_2 N = 39.2$ bits $\approx 52H_b(\frac{1}{4})$ bits.
b) $H(card) = H(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}) = \log_2 4 = 2$. So $H(deal) \approx 52H(card) = 104$. Allocate two bits to each of the 52 cards. Set them equal to 01 if the card is in the first hand, 10 if the card is in the second hand, 11 if the card is in the third hand, or 00 if the card is in the fourth hand.
c) Think of this as two selection processes. First choose 26 cards to be in one of the two hands, then choose 13 of these cards to be in a particular hand. $H \approx 52H_b(\frac{1}{2}) + 26H_b(\frac{1}{2}) = 78$ First allocate one bit to each card, setting it equal to 0 if the card is not in eiher of the two hands or a 1 if the card is in one of the hands. Then we can allocate the remaining 26 bits to the cards that are in one of the two hands. For these cards, let it equal 0 if the card is in the first hand and a 1 if the card is in the second hand.

8. a) There are 25 possible outcomes (Ball 1 is light, Ball 1 is heavy, . . . , ball 12 is heavy, all balls are normal weight). So the initial uncertainty is at most $\log_2 25$ bits. Each weighing results in one of three outcomes (heavy, light, equal). If each weighing is arranged so that all three of these outcomes are equiprobable, then we should be able to remove $3\log_2 3 = \log_2 27$ bits of uncertainty. So it may be possible.
b) Consider the following matrix:

$$\begin{pmatrix} 0 & 0 & -1 & 0 & -1 & -1 & 0 & -1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & -1 & 0 & -1 & 1 & 0 & -1 & 1 & -1 & 1 \\ 1 & 0 & 0 & -1 & -1 & 0 & -1 & 1 & 0 & -1 & 1 & 1 \end{pmatrix}$$

Each row represents a weighing and each column represents a ball. A -1 in the (i,j) entry means that in the ith weighing, the jth ball is the left pan, 1 means it is in the right pan, 0 means that the ball is left out of the ith weighing. If the three outcomes match a column, then the ball corresponding to that column is heavy. If the three outcomes mtch the inverse of some column, then the corresponding ball is light. If all weighings come out equal, then all balls are the same weight.
c) There are 13 potentially bad balls, which means that 27 outcomes are possible, so this problem may still be solvable. Let the good ball be number 14. As before, we use a ternary parity check matrix, with the same strategy as above.

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & -1 & -1 & -1 & 1 & -1 & 1 & -1 \\ 0 & 1 & 1 & 1 & -1 & -1 & -1 & 0 & 0 & 0 & 1 & -1 & 1 & -1 \\ 1 & -1 & 0 & 1 & -1 & 0 & 1 & 1 & 0 & -1 & -1 & 0 & 1 & -1 \end{pmatrix}$$

d) Now there are $29 > 27$ possible outcomes, so this problem is not solvable in 13 weighings.
e) We can either weigh 4 or 5 balls in each pan. If we weigh 4 against 4, and they are even, then we have 5 balls remaning. The uncertainty due to 5 balls is $\log_2 11$, but we can only remove $2\log_2 3 = \log_2 9$ bits of uncertainty in the 2 remaining weighings. If we weigh 5 against 5, and they are uneven, then if we are in the same position as before, and we cannot remove more than $\log_2 9$ bits of uncertainty in the 2 remaining weighings, so the problem is not solvable.