ECE 340: PROBABILISTIC METHODS IN ENGINEERING

SOLUTIONS TO HOMEWORK #1

2.1 Solution

a) The sample space consists of the twelve hours:
   \[ S = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\} \]

b) \[ A = \{1, 2, 3, 4\} \quad B = \{2, 3, 4, 5, 6, 7, 8\} \quad D = \{1, 3, 5, 7, 9, 11\} \]

   \[ A \cap B = \{3\} \]
   \[ A^C \cap B = \{5, 6, 7, 8\} \]
   \[ A \cup (B \cap D) = \{1, 2, 3, 4\} \cup (\{2, 3, 4, 5, 6, 7, 8\} \cap \{2, 4, 6, 8, 10, 12\}) = \{1, 2, 3, 4, 6, 8\} \]
   \[ (A \cup B) \cap D = \{2, 4, 6, 8\} \]

2.2 Solution

a) Sample space:
   \[ S = \{(1,1),(1,2),(1,3),(1,4),(1,5),(1,6),\]
   \[ (2,1),(2,2),(2,3),(2,4),(2,5),(2,6),\]
   \[ (3,1),(3,2),(3,3),(3,4),(3,5),(3,6),\]
   \[ (4,1),(4,2),(4,3),(4,4),(4,5),(4,6),\]
   \[ (5,1),(5,2),(5,3),(5,4),(5,5),(5,6),\]
   \[ (6,1),(6,2),(6,3),(6,4),(6,5),(6,6)\}\]

b)
   \[ A = \{(1,1),(2,1),(2,2),(3,1),(3,2),(3,3),(4,1),(4,2),(4,3),(4,4),(5,1),(5,2),(5,3),(5,4),(5,5),(6,1),(6,2),(6,3),(6,4),(6,5),(6,6)\}\]

Homework #1 Solutions
c) \( B = \{(6,1), (6,2), (6,3), (6,4), (6,5), (6,6)\} \)

d) \( B \) is a subset of \( A \) so when \( B \) occurs then \( A \) also occurs, thus \( B \) implies \( A \). But \( A \) does not imply \( B \).

e) 
\[
A \cap B^C = \{(1,1) \\
(2,1) (2,2) \\
(3,1) (3,2) (3,3) \\
(4,1) (4,2) (4,3) (4,4) \\
(5,1) (5,2) (5,3) (5,4) (5,5) \}
\]
The event is “number of dots in first toss is not 6 and not less than the number of dots in the second toss.”

f) \( C = \{(1,3), (2,4), (3,1), (3,5), (4,2), (4,6), (5,3), (6,4)\} \).
Hence, \( A \cap C = \{(3,1), (4,2), (5,3), (6,4)\} \)

2.5 Solution

a) Each testing of a pen has two possible outcomes: “pen good” \( g \) or “pen bad” \( b \). The experiment consists of testing pens until a good pen is found. Therefore, each outcome of the experiment consists of a string of “b’s” ended by a “g.” We assume that each pen is not put back in the drawer after being tested. Thus, \( S = \{g, bg, bbg, bbgb, bbbgg\} \)

b) We now simply record the number of pens tested, so \( S = \{1, 2, 3, 4, 5\} \)

c) \( S = \{gg, bgg, gbg, bbgb, bbgb, ggbgb, bbgb, bbbgg, gbbgb, bgbbg, bbgb, bbbgb, bbgb, bbbgb, bbgbg, bbbgg\} \)

d) \( S = \{2, 3, 4, 5, 6\} \)
2.9 If we sketch the events A and B we see that B=A U C. We also see that the intervals corresponding to A and C have no points in common, so A ∩ C = ∅.

We also see that (r,s]= (r,∞) \cap (-∞,s] = (-∞,r]\cap (-∞,s]; that is C=A^C \cap B. If we sketch the events A and B we see that B=A U C. More formally,

\[ A \cup C = A \cup (A^C \cap B) = (A \cup A^C) \cap (A \cup B) = \emptyset \cap (A \cup B) = A \cup B = B. \]

Or,

\[ B = B \cap \Omega = B \cap (A \cup A^C) = (B \cap A) \cup (B \cap A^C) = (B \cap A) \cup C = A \cup C. \]

To see that A \cap C = ∅, we do the following

\[ A \cap C = A \cap (A^C \cap B) = (A \cap A^C) \cap B = \emptyset \cap B = ∅. \]

2.12 Solution:

Note that if A \cup B = A, then B ⊂ A.

To prove this by contradiction: if b is an element of B but not an element of A (contrary to the conclusion B ⊂ A), then necessarily A is a proper subset of A \cup B (this means A \cup B has at least one element that is not in A), which is a contradiction to the assumption that A \cup B = A. So B ⊂ A. (*)

Similarly, we can prove if A \cap B = A then A ⊂ B. To prove this, note that if A has a member, say a, that is not in B, then clearly, a cannot be in A \cap B either, which implies that A \cap B ≠ A, a contradiction. So A ⊂ B. (**)

Now we know that the definition of set equality tells us that A=B if and only if (also written as “iff”) A ⊂ B and B ⊂ A. Then from (*) and (**), we conclude that that A=B.

2.14 Solution:

a) \((A \cap B^c \cap C^c) \cup (B \cap A^c \cap C^c) \cup (C \cap A^c \cap B^c) = E_a; \)

b) \((A \cap B \cap C^c) \cup (B \cap A \cap C) \cup (C \cap A \cap B^c) = E_b; \)

c) \(A \cup B \cup C, \) or equivalently, \(E_a \cup E_b \cup (A \cap B \cap C); \)

d) \(E_b \cup (A \cap B \cap C); \)

e) \(A^c \cap B^c \cap C^c, \) or equivalently \((A \cup B \cup C)^c. \)
MATLAB assignment

a) Important comments are provided in the following

```matlab
% START OF PROGRAM
% game.m
% CREATED BY M. HAYAT ON 8/30/99 FOR ECE211
% THIS PROGRAM SIMULATES THE GAME STRATEGY PROBLEM
% DISCUSSED IN CLASS
% NOTE: THE PERCENT SYMBOL % IS USED IN MATLAB TO
% COMMENT OUT THE TEXT THAT FOLLOWS IT
clear
close(figure(1))
Max=input('What is the maximum number of trials?')
f=0;
sf2=0;
for T=1:Max;
x=rand; % generate a number x that is uniformly distributed between 0 and 1.
t1=1/3; t2=2/3;
s=zeros(1,3);
if (x<t1) % 33% of the time, put it behind door 1
    s(1)=1;
else if (x<t2) % 33% of the time, put it behind door 2
    s(2)=1;
else % 33% of the time, put it behind door 3
    s(3)=1;
end
y=rand; % generate player's choice
if (y<t1) i=1; % 33% of the time, guess door 1
else if (y<t2) i=2; % 33% of the time, guess door 2
else i=3; % 33% of the time, guess door 3
end
f=f+s(i); % if the guess is correct, then count 1
mean=f/T; % find the updated mean
p(T)=mean; % put the mean to the vector p as a function of T = 1:Max

% now consider the switch policy
x=rand;
s=zeros(1,3);
if (x<t1)
    s(1)=1;
elseif (x<t2)
    s(2)=1;
else
    s(3)=1;
end
k=find(s);
y=rand;
if (y<t1) i=1;
elseif (y<t2) i=2;
else i=3;
end
if (k == i) % if the pick is correct
    if (k == 1) % and if the prize is behind door 1
        j=2; % the player switches to door 2
    else % if the prize is not behind door 1
        j = 1; % the player switches to door 1
end
```

```
else if the initial pick is not correct
    j=k; % then the player switch to the correct door
end
sf2=sf2+s(j); % is the choice j is correct, add the counter
smean=sf2/T; % find the updated mean
p2(T)=smean;
end
TT=[1:1:Max];
plot(TT,p,'-','TT,p2,--')
grid xlabel('NUMBER OF TRIALS')
ylabel('WINNING FREQUENCY')
legend('DON''T SWITCH POLICY','SWITCH POLICY')
axis([1 Max 0 1]);

%END OF PROGRAM

b) Estimates of the probability of winning as a function of the number of trials

<table>
<thead>
<tr>
<th>Number of Trials</th>
<th>Don’t switch</th>
<th>Switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.5</td>
<td>0.8</td>
</tr>
<tr>
<td>100</td>
<td>0.41</td>
<td>0.7</td>
</tr>
<tr>
<td>1000</td>
<td>0.303</td>
<td>0.662</td>
</tr>
</tbody>
</table>

As the number of trials increases the winning probability of winning for the “don’t switch policy” approaches 1/3 and for the “switch policy” it approaches 2/3.
c) To simulate the fact that the dealer is biased in placing the prize behind door #1 50% of the time, the lines of code shown in **bold** need to be modified or added.

```matlab
% START OF PROGRAM
% game.m
% CREATED BY M. HAYAT ON 8/30/99 FOR ECE211
% % THIS PROGRAM SIMULATES THE GAME STRATEGY PROBLEM
% % DISCUSSED IN CLASS
% % NOTE: THE PERCENT SYMBOL % IS USED IN MATLAB TO
% % COMMENT OUT THE TEXT THAT FOLLOWS IT
clear
close(figure(1))
Max=input('What is the maximum number of trials?')
f=0;
sf2=0;
for T=1:Max;
x=rand; % generate a number x that is uniformly distributed between 0 and 1.
t1=1/2; t2=1/4;
s=zeros(1,3);
if (x<t1)
s(1)=1; % 33% of the time, put it behind door 1
elseif (x<t2)
s(2)=1; % 33% of the time, put it behind door 2
else
s(3)=1; % 33% of the time, put it behind door 3
end
t1=1/3;
t2=2/3;
y=rand; % generate player's choice
if (y<t1)
i=1; % 33% of the time, guess door 1
elseif (y<t2)
i=2; % 33% of the time, guess door 2
else
i=3; % 33% of the time, guess door 3
end
f=f+s(i); % if the guess is correct, then count 1
mean=f/T; % find the updated mean
p(T)=mean; % put the mean to the vector p as a function of T = 1:Max
% now consider the switch policy
```
x=rand;
s=zeros(1,3);
if (x<t1)
  s(1)=1;
elseif (x<t2)
  s(2)=1;
else
  s(3)=1;
end
k=find(s);
t1=1/3;
t2=2/3;
y=rand;
if (y<t1)
i=1;
elseif (y<t2)
i=2;
else
i=3;
end
if (k == i) % if the pick is correct
  if (k == 1) % and if the prize is behind door 1
    j=2;  % the player switches to door 2
  elseif if the prize is not behind door 1
    j = 1;  % the player switches to door 1
  end
else  % if the initial pick is not correct
  j=k;  % then the player switch to the correct door
end
sf2=sf2+s(j);  % is the choice j is correct, add the counter
smean=sf2/T;  % find the updated mean
p2(T)=smean;
end

t=[1:1:Max];
plot(t,p,'-');tt,p2,'--')
grid
xlabel('NUMBER OF TRIALS')
ylabel('WINNING FREQUENCY')
legend('DON''T SWITCH POLICY','SWITCH POLICY')
axis([1 Max 0 1]);

%END OF PROGRAM

The new winning frequencies and plots are (for example):

<table>
<thead>
<tr>
<th>Number of Trials</th>
<th>Don’t switch</th>
<th>Switch</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.30</td>
<td>0.70</td>
</tr>
<tr>
<td>100</td>
<td>0.43</td>
<td>0.63</td>
</tr>
<tr>
<td>1000</td>
<td>0.336</td>
<td>0.669</td>
</tr>
</tbody>
</table>
d) From the results of part c it seems that the winning probabilities are not dependent on
the distribution used to place the prize initially behind either. Even the prize is hidden in
any one of the doors 100%, the player is not assumed to have this information, this
leads to the same probability of winning either policy.