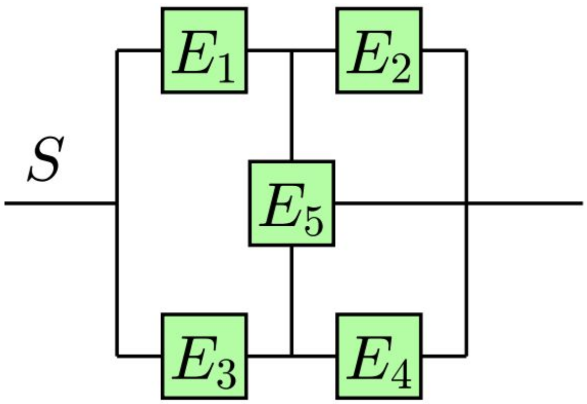
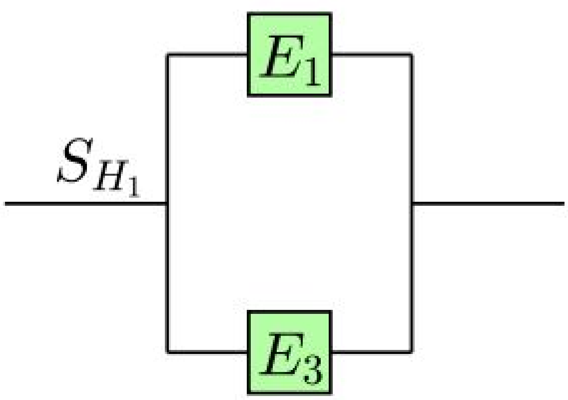
1. (a) In the below circuit, to find the probability for the circuit to be operational at time t, we'll analyze the circuit by splitting the in two cases



Case 1: Let H_1 be the event that E_5 works.

Our circuit is now simplified to:

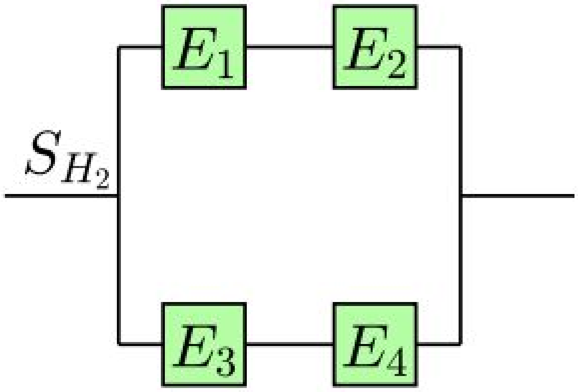


Probability of case 1 happening is:
$P(H_1) = P(E_5) = e^{-t}$

$P(S|H_1) = P(E_1 \cup E_3)$ {From Case 1 figure}
$= P(E_1) + P(E_3) - P(E_1 \cap E_3)$
$= P(E_1) + P(E_3) - P(E_1) * P(E_3)$ {Since, E_1 and E_3 are independent $P(E_1 \cap E_3) = P(E_1) * P(E_3)$}
$= e^{-t} + e^{-t/2} - (e^{-t} * e^{-t/2})$
$= e^{-t} + e^{-t/2} - e^{-3t/2}$

Case 2: Let H_2 be the event that E_5 does not work.

Our circuit is now simplified to:



Probability of case 2 happening is:
$P(H_2) = P(H_1^C) = 1 - P(H_1) = 1 - e^{-t}$

Since, E_1 and E_2 are in series,
the probability of the top component working
= probability of both E_1 and E_2 working
$= P(E_1 \cap E_2)$
$= P(E_1) * P(E_2)$
$= e^{-t} * e^{-2t}$
$= e^{-3t}$

Similarly, E_3 and E_4 are in series,
the probability of the bottom component working
= probability of both E_3 and E_4 working
$= P(E_3 \cap E_4)$
$= P(E_3) * P(E_4)$
$= e^{-t/2} * e^{-t/3}$
$= e^{-5t/6}$

$P(S|H_2) = P((E_1 \cap E_2) \cup (E_3 \cap E_4))$ {From Case 2 figure}
$= P(E_1 \cap E_2) + P(E_3 \cap E_4) - P((E_1 \cap E_2)\ _1\cap\ (E_3 \cap E_4))$
$= P(E_1 \cap E_2) + P(E_3 \cap E_4) - P(E_1 \cap E_2) * P(E_3 \cap E_4)$ {Since, top component and bottom component work independently $P((E_1 \cap E_2)\ _1\cap\ (E_3 \cap E_4)) = P(E_1 \cap E_2) * P(E_3 \cap E_4)$}
$= e^{-3t} + e^{-5t/6} - (e^{-3t} * e^{-5t/6})$
$= e^{-3t} + e^{-5t/6} - e^{-23t/6}$

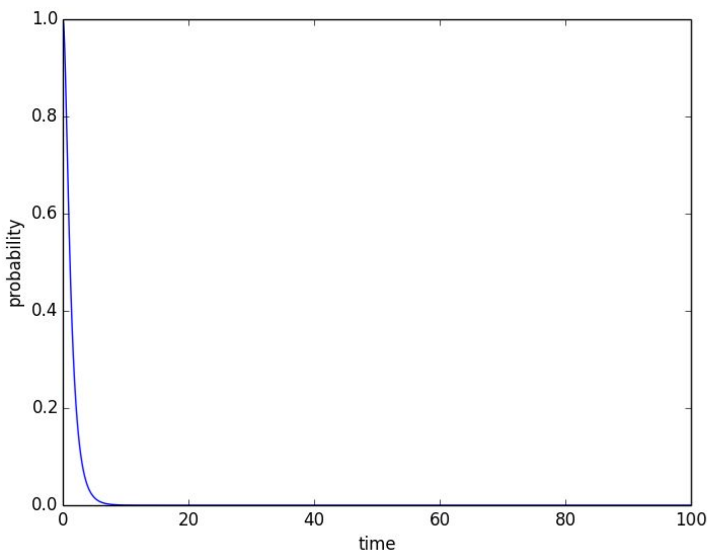
Combining Case 1 and Case 2 to find total probability of circuit S working:

$P(S) = P(S|H_1) * P(H_1) + P(S|H_2) * P(H_2)$
$= (e^{-t} + e^{-t/2} - e^{-3t/2}) * e^{-t} + (e^{-3t} + e^{-5t/6} - e^{-23t/6}) * (1 - e^{-t})$
$= e^{-2t} + e^{-3t/2} - e^{-5t/2} + e^{-3t} + e^{-5t/6} - e^{-23t/6} - e^{-4t} - e^{-11t/6} + e^{-29t/6}$

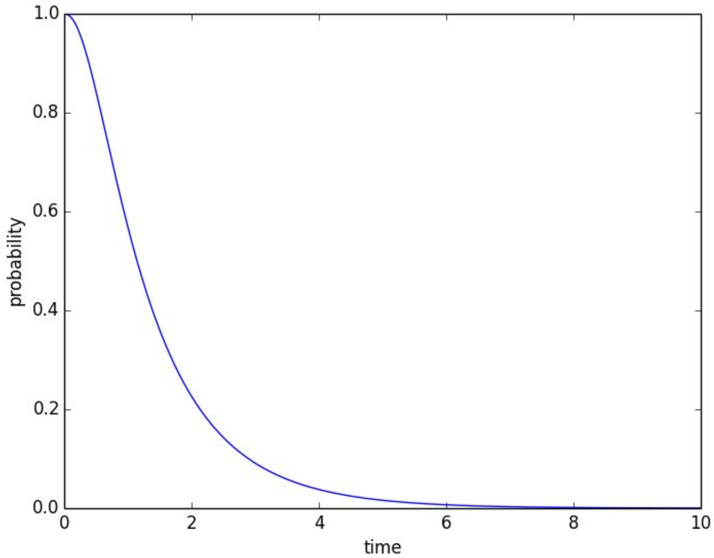
$= e^{-2t} + e^{-3t} + e^{-3t/2} + e^{-5t/6} + e^{-29t/6} - e^{-5t/2} - e^{-23t/6} - e^{-4t} - e^{-11t/6}$

Since, t cannot be negative as it is time, I have only plotted for positive values of t.

Plot: as t varies from 0-100 (Generated using Python. Code provided at the end of this assignment)
$p(t) = e^{-2t} + e^{-3t} + e^{-3t/2} + e^{-5t/6} + e^{-29t/6} - e^{-5t/2} - e^{-23t/6} - e^{-4t} - e^{-11t/6}$



Zooming Plot: as t varies from 0-10 (Generated using Python. Code provided at the end of this assignment)
$p(t) = e^{-2t} + e^{-3t} + e^{-3t/2} + e^{-5t/6} + e^{-29t/6} - e^{-5t/2} - e^{-23t/6} - e^{-4t} - e^{-11t/6}$



Probability at t=½: (Calculated using python code. Code provided in the Code section of this assignment)
P(1/2) = 0.843049099836

(b) Probability that E_5 was operational at time t, given that system was operational at that time is given by $P(H_1|S)$.

$P(H_1|S) = P(S|H_1) * P(H_1)/P(S)$

$= (e^{-t} + e^{-t/2} - e^{-3t/2}) * e^{-t}/(e^{-2t} + e^{-3t} + e^{-3t/2} + e^{-5t/6} + e^{-29t/6} - e^{-5t/2} - e^{-23t/6} - e^{-4t} - e^{-11t/6})$

$= (e^{-2t} + e^{-3t/2} - e^{-5t/2})/(e^{-2t} + e^{-3t} + e^{-3t/2} + e^{-5t/6} + e^{-29t/6} - e^{-5t/2} - e^{-23t/6} - e^{-4t} - e^{-11t/6})$

At t = ½ :

$P(H_1|S)$ = 0.656831490787 (Calculated using python code. Code provided in the Code section of this assignment)

2. Let C be the event that the product is conforming and C^c be the event that the product is non-conforming.
Let B_1 be the event that the product is selected from Batch 1.
Let B_2 be the event that the product is selected from Batch 2.

Given:
$P(C|B_1) = 1$
$P(C^c|B_1) = 0$
$P(C|B_2) = 0.8$
$P(C^c|B_2) = 0.2$
$P(B_1) = ½$
$P(B_2) = ½$
Using total probability theorem for finding the Probability of the product being confirming:
$P(C) = P(C|B_1) * P(B_1) + P(C|B_2) * P(B_2)$
$= 1*½ + 0.8 * ½$
$= 0.9$

Using Bayes theorem to calculate $P(B_1|C)$:
$P(B_1|C) = P(C|B_1) * P(B_1)/P(C)$
$= (½ * 1)/0.9$
$= 5/9$

Similarly, using Bayes theorem to calculate $P(B_2|C)$:
$P(B_2|C) = P(C|B_2) * P(B_2)/P(C)$
$= (0.8 * ½) / 0.9$
$= 4/9$

Probability that the second product, randomly selected from the same batch, is found non-conforming is given by P (C^c| first product chosen was conforming) and can be calculated by total probability formula.

Since, the first event has already occurred, our probability that B_1 or B_2 to be chosen has been updated.

Now,
P(B_1|first chosen product was conforming) = 5/9
P(B_2|first chosen product was conforming) = 4/9

P (C^c| first product chosen was conforming) = P(B_1|first product chosen was conforming) * P ($C^c|B_1$)
+ P(B_2|first product chosen was conforming) * P ($C^c|B_2$)
= 5/9 * 0 + 4/9 * 0.2
= 4/45
= 0.0888889 (approximated)

---

3. Let A_0 be the event that the actual classification of the item is 0.
Let A_1 be the event that the actual classification of the item is 1.
Let P_0 be the event that the predicted classification of the item is 1.
Let P_1 be the event that the predicted classification of the item is 1.

Following is the confusion matrix, that I obtained from the problem.

N = 120	Predicted 0 (P_0)	Predicted 1 (P_1)	
Actual 0 (A_0)	37	18	55
Actual 1 (A_1)	13	52	65
	50	70	

From above confusion matrix:

Probability of the classifier that it predicts classification as 1 when the actual classification is 1
$= P(P_1|A_1) = 52/65$

Probability of the classifier that it predicts classification as 1 when the actual classification is 0
$= P(P_1|A_0) = 18/55$

In the new population of items, we have:
Let $(NA)_0$ be the event that the actual classification of the item is 0 in the new population of items.
Let $(NA)_1$ be the event that the actual classification of the item is 1 in the new population of items.
Let $(NP)_0$ be the event that the predicted classification of the item is 0 in the new population of items.
Let $(NP)_1$ be the event that the predicted classification of the item is 1 in the new population of items.

$P((NA)_0) = 0.99$
$P((NA)_1) = 0.01$

$P((NP)_1|(NA)_1) = P(P_1|A_1) = 52/65$ {From the confusion matrix we know the probability of the classifier that it predicts classification as 1 when the actual classification is 1}

$P((NP)_1|(NA)_0) = P(P_1|A_0) = 18/55$ {From the confusion matrix we know the probability of the classifier that it predicts classification as 1 when the actual classification is 0}

We want to find out: $P((NA)_1|(NP)_1)$
Using Bayes formula:

$P((NA)_1|(NP)_1)$
$= P((NP)_1|(NA)_1) * P((NA)_1) / (P((NP)_1|(NA)_1) * P((NA)_1) + P((NP)_1|(NA)_0) * P((NA)_0))$
$= 52/65 * 0.01 / (52/65 * 0.01 + 18/55 * 0.99)$
$= 0.0241$ (approximated)

CODE

```
# Python code for plot as t varies from 0-10
import matplotlib.pyplot as plt
import numpy as np

# Create the vectors p and t
t = np.arange(0,10,0.0001)
p = np.exp(-2*t) + np.exp(-3*t) + np.exp(-3*t/2) + np.exp(-5*t/6) + np.exp(-29*t/6) - np.exp(-5*t/2) -
np.exp(-23*t/6) - np.exp(-4*t) - np.exp(-11*t/6)

# Create the plot
plt.plot(t,p)

plt.xlabel('time')
plt.ylabel('probability')

# Show the plot
plt.show()
```

```
# Python code for plot as t varies from 0-100
import matplotlib.pyplot as plt
import numpy as np

# Create the vectors p and t
t = np.arange(0,100,0.0001)
p = np.exp(-2*t) + np.exp(-3*t) + np.exp(-3*t/2) + np.exp(-5*t/6) + np.exp(-29*t/6) - np.exp(-5*t/2) -
np.exp(-23*t/6) - np.exp(-4*t) - np.exp(-11*t/6)

# Create the plot
plt.plot(t,p)

plt.xlabel('time')
plt.ylabel('probability')

# Show the plot
plt.show()
```

```
# Python code for finding probability P(S) at t=½

import matplotlib.pyplot as plt
import numpy as np

# Create the vectors p and t
t = np.arange(0,10,0.5)
p = np.exp(-2*t) + np.exp(-3*t) + np.exp(-3*t/2) + np.exp(-5*t/6) + np.exp(-29*t/6) - np.exp(-5*t/2) -
np.exp(-23*t/6) - np.exp(-4*t) - np.exp(-11*t/6)

pAtTisEqualToHalf = np.interp(0.5,t,p)

print(pAtTisEqualToHalf)
```

```
# Python code for finding probability P(H1| S) at t=½

import matplotlib.pyplot as plt
import numpy as np

# Create the vectors p and t
t = np.arange(0,10,0.5)
p = (np.exp(-2*t) + np.exp(-3*t/2) - np.exp(-5*t/2))/(np.exp(-2*t) + np.exp(-3*t) + np.exp(-3*t/2) + np.exp(-5*t/6) + np.exp(-29*t/6) - np.exp(-5*t/2) - np.exp(-23*t/6) - np.exp(-4*t) - np.exp(-11*t/6))

pAtTisEqualToHalf = np.interp(0.5,t,p)

print(pAtTisEqualToHalf)
```