Ma5

by Sigbjørn Fjelland

Date: '2022-11-04'

Due to issues with knit in r, and RCall package, the code wil be in Julia. I shal comment out so that it is fully understandable

Libraries in use

```
using Dates ,DataFrames ,Plots , PlutoUI , Random , LaTeXStrings ,
Statistics , LinearAlgebra , StatsPlots , Distributions
```

Problem 5.1 a)

```
n = 100
```

```
λ = 0.6
```

```
μ = 0.7
```

```
p = 0.46153846153846156
```

```
#Proportion Entering in to the system
p=λ/(λ+μ)
```

```
q = 0.5384615384615385
```

```
#Proportion being served
q=μ/(λ+μ)
```

```
jump (generic function with 2 methods)
```

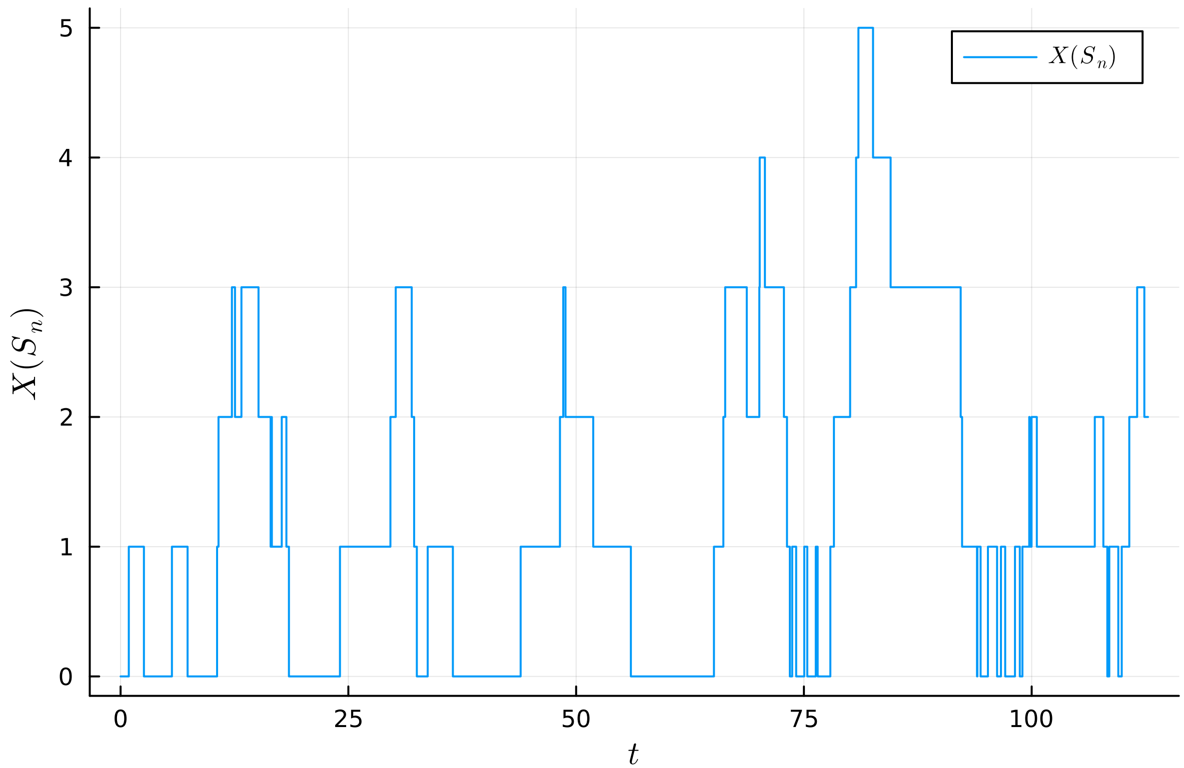
```
function jump(p,n, initialvalue=0)
        U = rand(n-1)
        queue = [initialvalue]                   #vector for storing each transition
        for i in 1:n-1
            if U[i] < p
                push!(queue,queue[i]+1)          #appends +1 to queue vector
            else
                push!(queue,max(0,queue[i]-1))   #appends -1 to queue vector with 0
                                                     as  floor
            end
        end
    return queue
end
```

```
jumptime (generic function with 1 method)
```

```
function jumptime(n)
    T=randexp(n-1)
    S=[0.00]
    for t in 1:(n-1)
        push!(S,S[t]+T[t]) #appends next cumulative sum (in lack of a cumsum function)
    end
    return S
end
```

```
[0.0, 0.629407, 0.902088, 2.56134, 2.79124, 5.63567, 7.35259, 10.5916, 10.7477, 12.2186, 1:
```

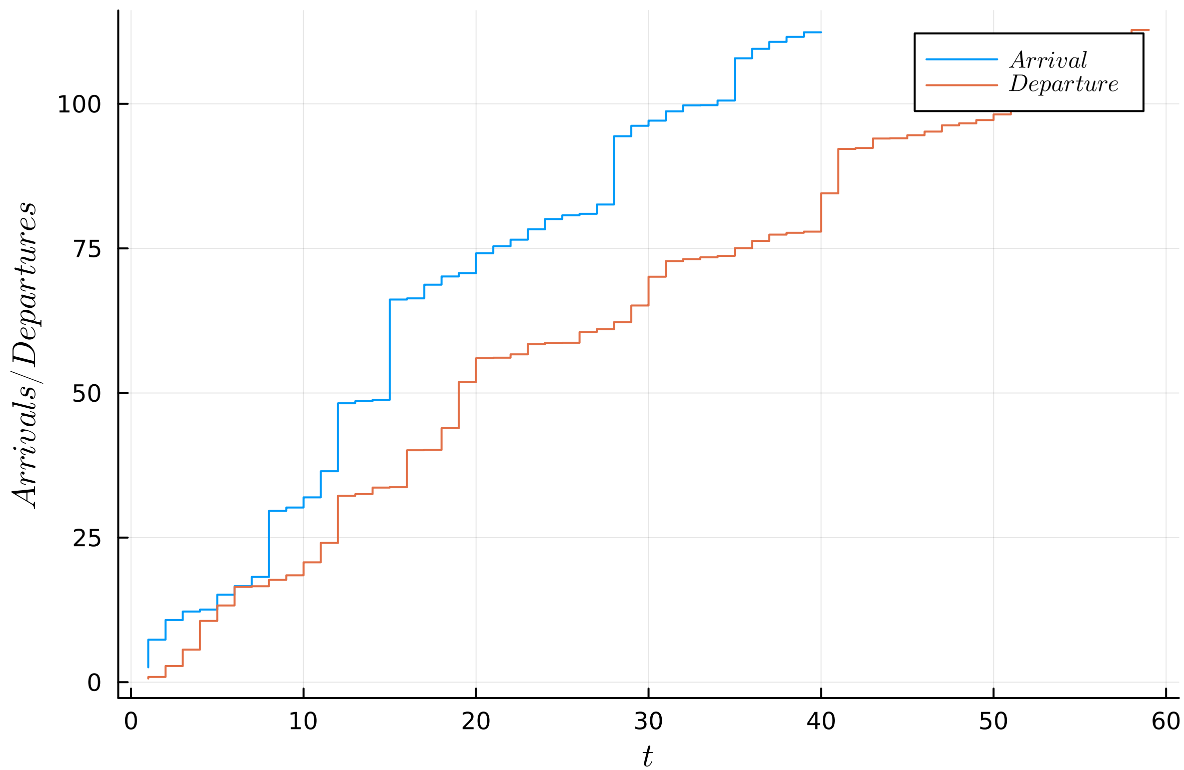
```
begin
Y_100=jump(p, 100)
S_100=jumptime(100)
end
```




```
plot(S_100, Y_100,
     titel = "simulation M/M/1 BD size n=100", label = L"X(S_n)",
     xlabel=L"t",  ylabel=L"X(S_n)", line = (:steppre))
```

Problem 5.1 b)

```
begin
arr = []
dep = []
for j in 2:n
    if Y_100[j]>Y_100[j-1]
        push!(arr, S_100[j])
    else
        push!(dep, S_100[j])
    end
end
end
```

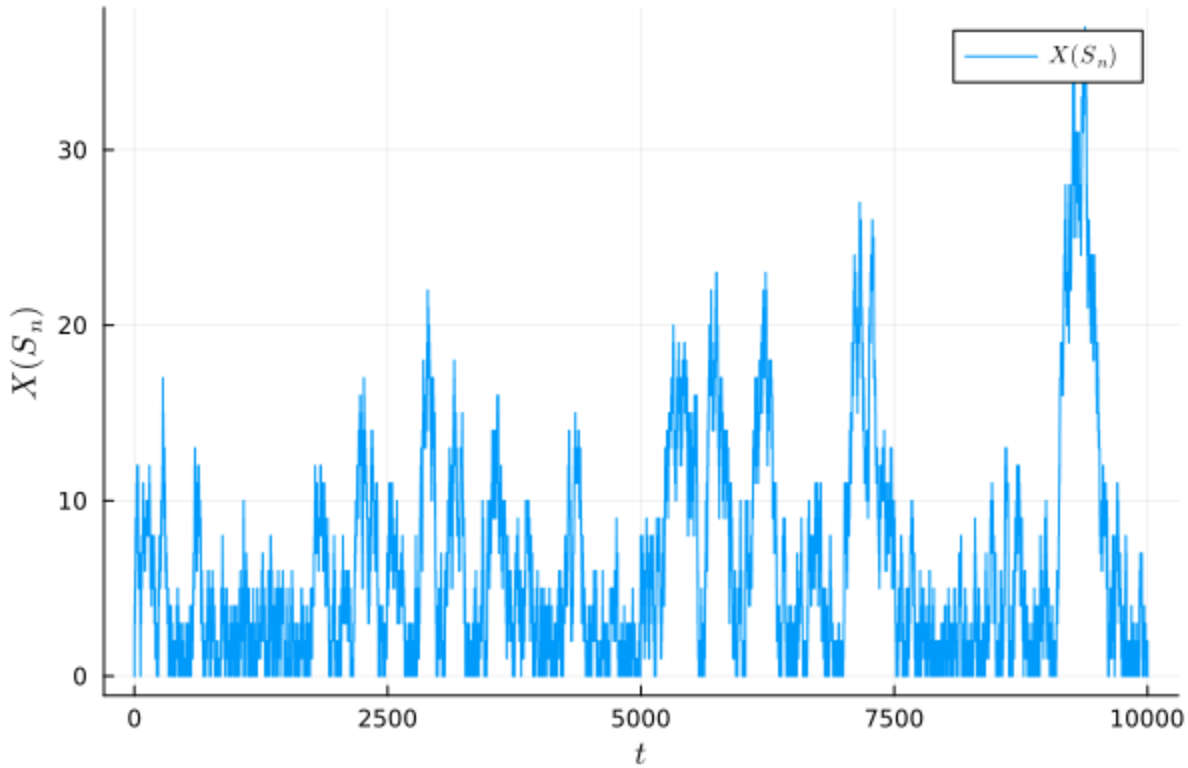



```
begin
plot(arr, label = L"Arrival",
     xlabel=L"t",  ylabel=L"Arrivals/Departures", linetype=:steppre)
plot!(dep, label = L"Departure", linetype=:steppre)
end
```

Problem 5.1 c)

```
[0.0, 0.124638, 0.206821, 0.753812, 0.820263, 0.979639, 2.64335, 2.99001, 4.38293, 5.1349,
```

```
begin
Y_10000=jump(p, 10000)
S_10000=jumptime(10000)
end
```

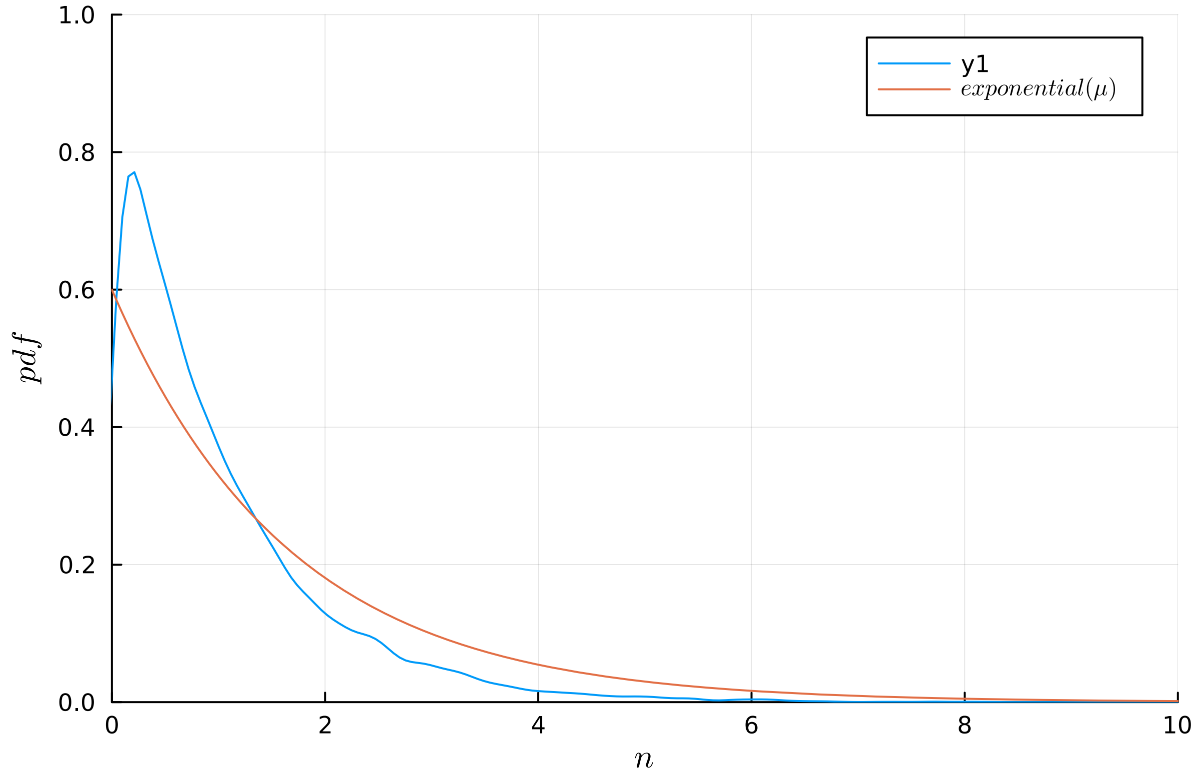



```
plot(S_10000, Y_10000,
     titel = "simulation M/M/1 BD size n=100", label = L"X(S_n)",
     xlabel=L"t",  ylabel=L"X(S_n)", linetype=:steppre)
```

```
holdtime (generic function with 1 method)
```

```julia
function holdtime(S)
T = []
for t in 2:length(S)
    push!(T, (S[t]-S[t-1]))
end
    return T
end

```




```julia
begin
    #density plot of holding times
    plot(density(holdtime(S_10000)),
        xlims= (0,10),
        ylims= (0,1),
        label = L"Holdtimes",
        titel = "Density of holdtimes",
        xlabel=L"n",
        ylabel=L"pdf")

        #Plot of distribution function
    plot!(Exponential((1/λ)),
        label = L"exponential(\mu)")
end
```

Problem 5.1 d)

```
π_0 = 0.1428571428571428
```
```
π_0 = 1-(λ/μ) #Stationary distribution
```

From the Embeded Markov chain we sort out the visits to state zero

```
S₀ =
  [1, 3, 4, 72, 73, 74, 240, 241, 243, 245, 246, 252, 356, 357, 358, 382, 383, 403, 407, 409,
```
```
S₀ = findall(Y_10000 .== 0) #argument of vector S₀=0 (simlar to which() in R)
```

From this we have to do the cumbersome way of extracting the holding time. This is due to som bad decition of keeping simulation of waiting time within the jumptime function..

```
begin
Times=holdtime(S_10000)
v = 0                          #Total time chain is in State zero
for i in S₀[1:length(S₀)-1]
      v = v + Times[i]
end
end
```

```
π̂₀ˣ = 0.13992245769115
```
```
π̂₀ˣ = (v+last(Times)) / last(S_10000)
```

```
π̂₀ʸ = 0.1355
```
```
π̂₀ʸ=length(S₀) / 10000
```

As we se the embedded chain focuses on jumps and increment while the chain it self is all about time and time spent in a state.

$$\hat{\pi}_0^X = \frac{\text{Time spent i 0}}{\text{Total time}}, \hat{\pi}_0^Y = \frac{\text{Jumps to 0}}{\text{Total number of jumps}}$$

a)

$$
\begin{tikzcd}
(0,0) \arrow[r, bend left, "\lambda_1"] \arrow[d, bend left, "\lambda_2"] & (1,0) \arrow[l, bend left, "\mu_1"] \arrow[d, bend left, "\lambda_2"] \\
(0,1) \arrow[u, bend left, "\mu_2"] \arrow[r, bend left, "\lambda_1"] & (1,1) \arrow[l, bend left, "\mu_1"] \arrow[u, bend left, "\mu_2"]
\end{tikzcd}
$$

b) <u>Infinitisimal</u> Matrix:

$$A = \begin{array}{cc} & \begin{array}{cccc} (0,0) & (1,0) & (0,1) & (1,1) \end{array} \\ \begin{array}{c} (0,0) \\ (1,0) \\ (0,1) \\ (1,1) \end{array} & \begin{bmatrix} -(\lambda_1+\lambda_2) & \lambda_1 & \lambda_2 & 0 \\ \mu_1 & -(\mu_1+\lambda_2) & 0 & \lambda_2 \\ \mu_2 & 0 & -(\lambda_1+\mu_2) & \lambda_1 \\ 0 & \mu_2 & \mu_1 & -(\mu_1, \mu_2) \end{bmatrix} \end{array}$$

c) <u>Transition Matrix</u>

$$
\mathbb{P} = \begin{array}{c} \\ (0,0) \\ (1,0) \\ (0,1) \\ (1,1) \end{array}
\begin{array}{c}
\begin{array}{cccc} (0,0) & (1,0) & (0,1) & (1,1) \end{array} \\
\begin{bmatrix}
0 & \left(\dfrac{\lambda_1}{\lambda_1+\lambda_2}\right) & \left(\dfrac{\lambda_2}{\lambda_1+\lambda_2}\right) & 0 \\[2ex]
\left(\dfrac{\mu_1}{\mu_1+\lambda_2}\right) & 0 & 0 & \left(\dfrac{\lambda_2}{\mu_1+\lambda_2}\right) \\[2ex]
\left(\dfrac{\mu_2}{\lambda_1+\mu_2}\right) & 0 & 0 & \left(\dfrac{\lambda_1}{\lambda_1+\mu_2}\right) \\[2ex]
0 & \dfrac{\mu_2}{\mu_1+\mu_2} & \dfrac{\mu_1}{\mu_1+\mu_2} & 0
\end{bmatrix}
\end{array}
$$

d) To be a "birth-deth" Process the chain can have only have only up/Down to neighbouring States...

e) $\pi^X A = 0$

$$X(t) = (X_1(t), X_2(t))$$

Where $X_1(t)$ and $X_2(t)$

are independent

$$\Rightarrow \quad X_1(t): \; 0 \underset{\mu_1}{\overset{\lambda_1}{\rightleftarrows}} 1 \qquad\qquad X_2(t): \; 0 \underset{\mu_2}{\overset{\lambda_2}{\rightleftarrows}} 1$$

generally $A = \begin{bmatrix} -\lambda & \lambda \\ \mu & -\mu \end{bmatrix}$

$$\tilde{\pi}^{x} A = 0$$

hence $\tilde{\pi}_0 = \dfrac{\mu}{\mu+\lambda}$, $\tilde{\pi}_1 = \dfrac{\lambda}{\mu+\lambda}$

$\tilde{\pi}^1_0 = \dfrac{\mu_1}{\mu_1+\lambda_1}$, $\tilde{\pi}^1_1 = \dfrac{\lambda}{\mu+\lambda}$

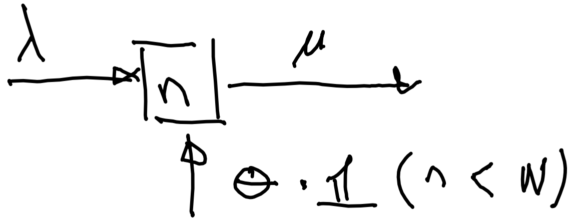
Same apply for $\tilde{\pi}^2_0$, $\tilde{\pi}^2_1$

$$\tilde{\pi} = \left(\tilde{\pi}^1_1 \cdot \tilde{\pi}^2_1,\ \tilde{\pi}^1_1 \tilde{\pi}^2_2,\ \tilde{\pi}^1_2 \tilde{\pi}^2_1,\ \tilde{\pi}^1_2 \tilde{\pi}^2_2\right)$$

$$= \left(\frac{\lambda_1}{\mu_1+\lambda_1},\ \frac{\mu_1}{\mu_1+\lambda_1}\right) \otimes \left(\frac{\lambda_2}{\mu_2+\lambda_2},\ \frac{\mu_2}{\mu_2+\lambda_2}\right)$$

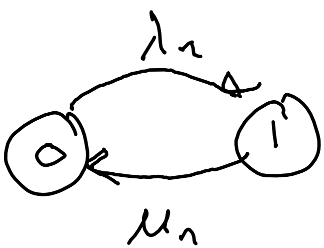
Tensor prod.

a) <u>Population :</u>



$$\lambda_n = \lambda n + \theta \mathbb{1}(n < N)$$

$$\mu_n = \mu \cdot n$$



b) $N=3$, $\lambda=\theta=1$, $\mu=2$

$X(t) \geq N \Rightarrow \mathbb{1}(n \leq N) = 0 \Rightarrow \theta = 0$

since we look at whats above

Immigration $\pi_0 = \pi_N$

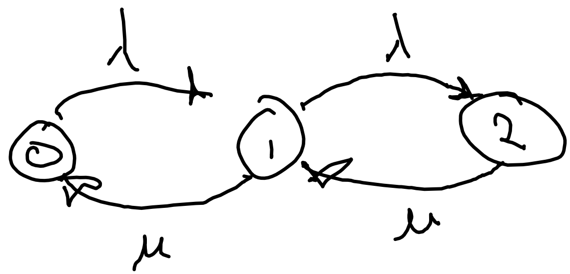
$$\pi_{N+k} = \frac{\lambda_{N+k-1}}{\mu_{N+k}} \pi_{N+k-1}$$

$$= \prod_{j=1}^{k} \frac{\lambda_{N+j-1}}{\mu_{N+j}} \pi_N = \prod_{j=1}^{k} \frac{(N+j-1)\lambda}{(N+j)\mu} \pi_N$$

Problem 5.4

$\lambda = 3\ hr^{-1}$ $\qquad \frac{1}{\mu} = \frac{1}{4} \Rightarrow \mu = 4$

a)



$S = \{0, 1, 2\}$

b)

This is a finite BD since there is a finite "S" and there is at most two directions out from each State

c) Avrage customers in shop:

$$\pi_0 = \left(\sum_{k=0}^{2} \frac{\prod_{i=0}^{n-1} \lambda}{\prod_{i=1}^{n} \mu} \right)^{-1} = \left(\sum_{k=0}^{2} \left(\frac{3}{4}\right)^k \right)^{-1}$$

$$= \left(1 + \left(\frac{3}{4}\right)^1 + \left(\frac{3}{4}\right)^2\right)^{-1} = \underline{\frac{16}{37}}$$

$$\pi_1 = \frac{3}{4} \cdot \frac{16}{37} = \underline{\frac{12}{37}}$$

$$\pi_2 = \left(\frac{3}{4}\right)^2 \cdot \frac{16}{37} = \frac{9}{37}$$

$$E[X(t)] \quad \pi_1 \cdot 1 + \pi_2 \cdot 2 = \frac{12}{37} + 2 \cdot \frac{9}{37} = \underline{0{,}81}$$

for all practicalreasons $0{,}81 \approx 1$ since 87% of a person is hard to find.

d)

$$\frac{\text{Entering rate}}{\text{arrival rate}} = \frac{\lambda \tilde{\pi}_0 + \lambda \tilde{\pi}_1}{\lambda} = \tilde{\pi}_0 + \tilde{\pi}_1$$

$$= \frac{16}{37} + \frac{12}{37} = \frac{28}{37} = \underline{0{,}7567}$$

e) doubling the work pace:

$\mu' = 2\mu = \underline{8}$

$\tilde{\pi}_0 = \left(1 + \frac{3}{8} + \left(\frac{3}{8}\right)^2\right)^{-1} \approx \underline{0{,}659}$

$\tilde{\pi}_1 = \tilde{\pi}_0 \cdot \frac{3}{8} \approx \underline{0{,}247}$

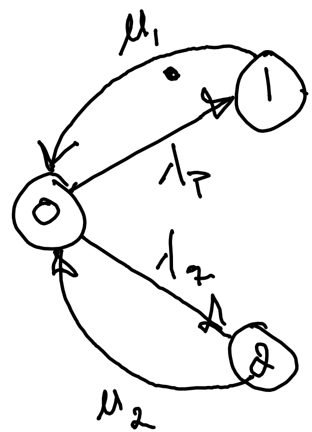
$\tilde{\pi}_1 + \tilde{\pi}_0 = 0{,}66 + 0{,}25 \approx \underline{0{,}9}$

Efficiency has increased from 0,75 to 0,9

Earnings improved by $\frac{0{,}9 - 0{,}75}{0{,}75} \approx 0{,}2$

Problem 5.5

a)



$\lambda_p = p\lambda \qquad \lambda_q = \lambda(1-p) = \lambda q$

$\boxed{q = 1 - p}$

Statespace of "Pacman" to the left is

$S = \{1, 0, 2\}$

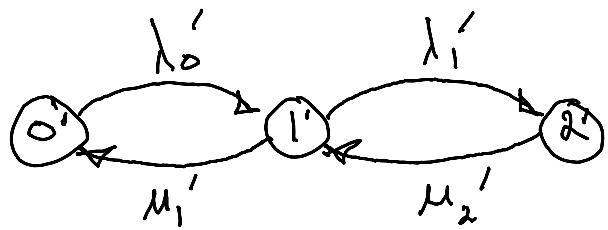
Wich can be denoted

$S' = \{0' = 1,\ 1' = 0,\ 2' = 2\}$

$\lambda_0' = \mu_1, \quad \lambda_1' = \lambda q,$

$\mu_1' = \lambda \cdot p, \quad \mu_2' = \mu_2$

This gives following chain (BD)



$$r_1 = p \frac{\lambda}{\mu_1} \qquad r_2 = q \frac{\lambda}{\mu_2} \qquad r_0 = 1$$

b)

$$
A = \begin{array}{c|ccc}
 & 0' & 1' & 2' \\
\hline
0' & -\lambda_0' & \mu_1' & 0 \\
1' & \lambda_0' & -(\lambda_1' + \mu_1') & \mu_2' \\
2' & \lambda_1' & 0 & -\mu_2'
\end{array}
\quad 0' \leftrightarrow 1' \quad \sim \quad
\begin{bmatrix}
\mu_1 & -(\lambda q + \lambda p) & \mu_2 \\
-\mu_1 & \lambda p & 0 \\
\lambda q & 0 & -\mu_2
\end{bmatrix}
$$

Subst. back original state

$$
\pi A = \begin{cases}
\pi_0 \mu_1 - \pi_1 \lambda \overbrace{(p+q)}^{=1} + \pi_2 \mu_2 = 0 \\
-\pi_0 \mu_1 + \pi_1 \lambda p = 0 \\
\pi_0 \lambda q - \pi_2 \mu_2 = 0
\end{cases}
$$

in addition $\sum \pi_i = 1$

$$\text{I} \quad \tilde{\pi}_0 \mu_1 + \tilde{\pi}_2 \mu_2 = \tilde{\pi}_1 \lambda$$

$$\text{II} \quad \tilde{\pi}_1 \lambda p = \tilde{\pi}_0 \mu_1$$

$$\text{III} \quad \tilde{\pi}_1 \lambda q = \tilde{\pi}_0 \mu_2$$

$$\text{IV} \quad \tilde{\pi}_0 + \tilde{\pi}_1 + \tilde{\pi}_2 = 1$$

$$(\text{II}) \Rightarrow \quad \tilde{\pi}_1 = p \frac{\lambda}{\mu_1} \tilde{\pi}_0 \quad , \quad \tilde{\pi}_1 = q \frac{\lambda}{\mu_2} \tilde{\pi}_0$$

$$1 = \tilde{\pi}_0 \left(1 + p \frac{\lambda}{\mu_1} + q \frac{\lambda}{\mu_2} \right)$$

$$\Rightarrow \quad \pi_0 = \frac{1}{1 + p\frac{\lambda}{\mu_1} + q\frac{\lambda}{\mu_2}} = \frac{1}{\left(\frac{\mu_1\mu_2 + p\lambda + q\lambda}{\mu_1\mu_2}\right)}$$

$$= \underline{\left(\frac{\mu_1\mu_2}{\mu_1\mu_2 + \lambda}\right)}$$

c)

The proportion of down time on Machine #1

$$\pi_1 = P \frac{\lambda}{\cancel{\mu_1}} \cdot \frac{\cancel{\mu_1} \mu_2}{\mu_1 \mu_2 + \lambda} = \frac{\lambda \mu_2}{\mu_1 \mu_2 + \lambda} \cdot P$$

d)

The proportion of time Machine #2 is down.

$$\tilde{\pi}_2 = 1 - \left(\frac{\lambda \mu_2}{\mu_1 \mu_2 + \lambda} \cdot p + \frac{\mu_1 \mu_2}{\mu_1 \mu_2 + \lambda} \right)$$

$$= \frac{\mu_1 \mu_2 + \lambda - p \cdot \lambda \mu_2 - \mu_1 \mu_2}{\mu_1 \mu_2 + \lambda}$$

$$= \frac{\mu_2 \left(\cancel{\mu_1} + \lambda (1-p) - \cancel{\mu_1} \right)}{\mu_1 \mu_2 + \lambda}$$

$$= q \left(\frac{\mu_2 \lambda}{\mu_1 \mu_2 + \lambda} \right)$$

e) Proportion of time both are up and running are π_0 (b).

f) $\tilde{\pi}_1 = \frac{\lambda \mu_2}{\mu_1 \mu_2 + \lambda} \cdot P = \frac{0{,}01 \cdot 0{,}1}{1 \cdot 0{,}1 + 0{,}01} \cdot 0{,}9 \approx \frac{9}{1100}$

hmm m .. somthing is wrong. !