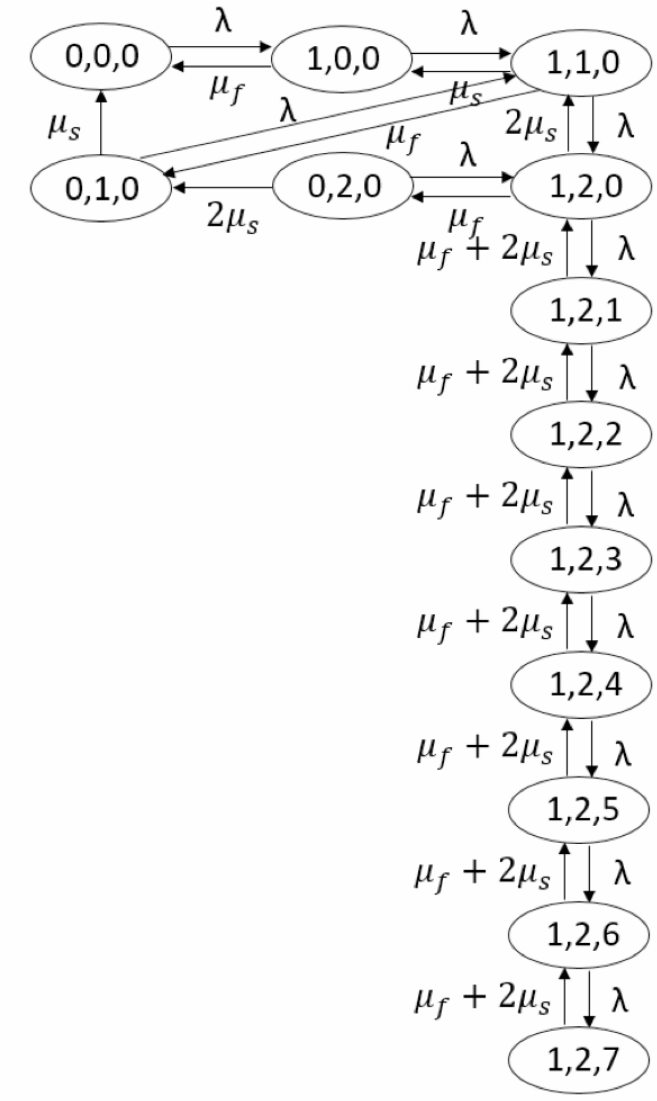
**Solution to Homework #2**
**CS5214 Modeling and Evaluation Fall 2023**

---

Problem #1: The Markov model is as follows:




**Source Code:**

```
bind
*customer arrival rate
lambda 15
muf 5
mus 3
end

markov M/M/3/10
* arrival
000 100 lambda
```

```
100 110 lambda
110 120 lambda
120 121 lambda
121 122 lambda
122 123 lambda
123 124 lambda
124 125 lambda
125 126 lambda
126 127 lambda
010 110 lambda
020 120 lambda
* departure
100 000 muf
110 100 mus
110 010 muf
120 110 2*mus
121 120 muf+2*mus
122 121 muf+2*mus
123 122 muf+2*mus
124 123 muf+2*mus
125 124 muf+2*mus
126 125 muf+2*mus
127 126 muf+2*mus
120 020 muf
020 010 2*mus
010 000 mus
end

format 5
echo ----------------------------------------
echo Q1: customer turned-away probability
echo ----------------------------------------
expr prob (M/M/3/10,127)
echo ----------------------------------------
echo Q2: average number of customers waiting in the system
var waitingn prob(M/M/3/10,121)*1 + prob(M/M/3/10,122)*2 + prob (M/M/3/10,123)*3 +
prob(M/M/3/10,124)*4 + prob(M/M/3/10,125)*5 + prob(M/M/3/10,126)*6 +
prob(M/M/3/10,127)*7
expr prob(M/M/3/10,121)*1 + prob(M/M/3/10,122)*2 + prob (M/M/3/10,123)*3 +
prob(M/M/3/10,124)*4 + prob(M/M/3/10,125)*5 + prob(M/M/3/10,126)*6 +
prob(M/M/3/10,127)*7
var X lambda * (1-prob (M/M/3/10,127))
* waitingR = average waiting time per customer by applying Little's law R=n/X
var waitingR waitingn/X
echo ----------------------------------------
echo Q3: response time for customers served by the fast server
```

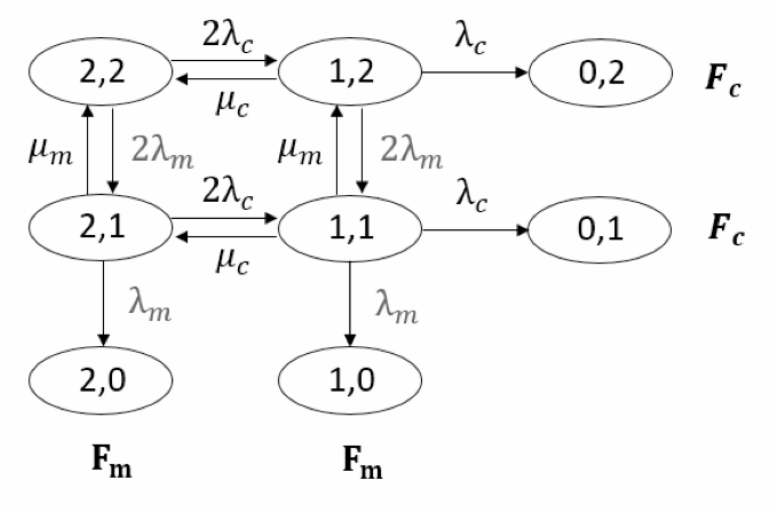
```
*response time = waiting time + service time at the fast server
expr waitingR + 1/muf
echo ----------------------------------------
echo Q4: response time for customers served by the slow servers
*response time = waiting time + service time at the slow server
expr waitingR + 1/mus
echo ----------------------------------------
echo Q5: throughput
expr lambda * (1-prob (M/M/3/10,127))
echo ----------------------------------------

end
```

**Output:**

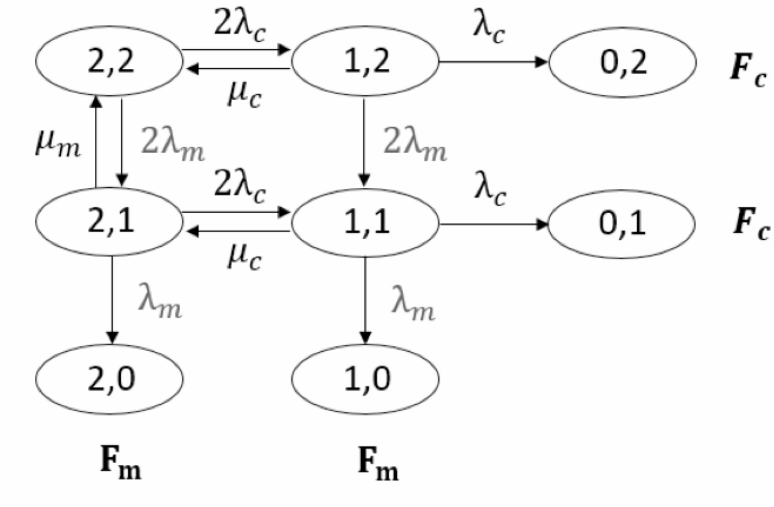
```
* ----------------------------------------
* Q1: customer turned-away probability
* ----------------------------------------
 prob (M/M/3/10,127):   2.80007e-01

* ----------------------------------------
* Q2: average number of customers waiting in the system
----------------------------------------

 prob(M/M/3/10,121)*1 + prob(M/M/3/10,122)*2 + prob (M/M/3/10,123)*3 +
prob(M/M/3/10,124)*4 + prob(M/M/3/10,125)*5 + prob(M/M/3/10,126)*6 +
prob(M/M/3/10,127)*7:   4.79195e+00

* ----------------------------------------
* Q3: response time for customers served by the fast server
----------------------------------------

 waitingR + 1/muf:   6.43703e-01

* ----------------------------------------
* Q4: response time for customers served by the slow servers
----------------------------------------

 waitingR + 1/mus:   7.77036e-01

* ----------------------------------------
* Q5: throughput
----------------------------------------

 lambda * (1-prob (M/M/3/10,127)):   1.07999e+01

* ----------------------------------------
```

Problem #2

The Markov models for cases (a)-(b) are as follows:
(a)




(b)



**Source Code:**

(a)

```
bind
lambdap 1/1000
lambdam 1/1200
mup 1/4
mum 1/2
end

markov hw2-2a
22 12 lambdap*2
12 02 lambdap
21 11 lambdap*2
11 01 lambdap
22 21 lambdam*2
12 11 lambdam*2
21 20 lambdam
11 10 lambdam
12 22 mup
11 21 mup
21 22 mum
11 12 mum
end

22 1.0
end

expr 1-value(500; hw2-2a)
end
```

(b)

```
bind
lambdap 1/1000
lambdam 1/1200
mup 1/4
mum 1/2
end

markov hw2-2b
22 12 lambdap*2
12 02 lambdap
21 11 lambdap*2
11 01 lambdap
22 21 lambdam*2
12 11 lambdam*2
```

```
21 20 lambdam
11 10 lambdam
12 22 mup
11 21 mup
21 22 mum
end

22 1.0
end

expr 1-value (500; hw2-2b)
end
```

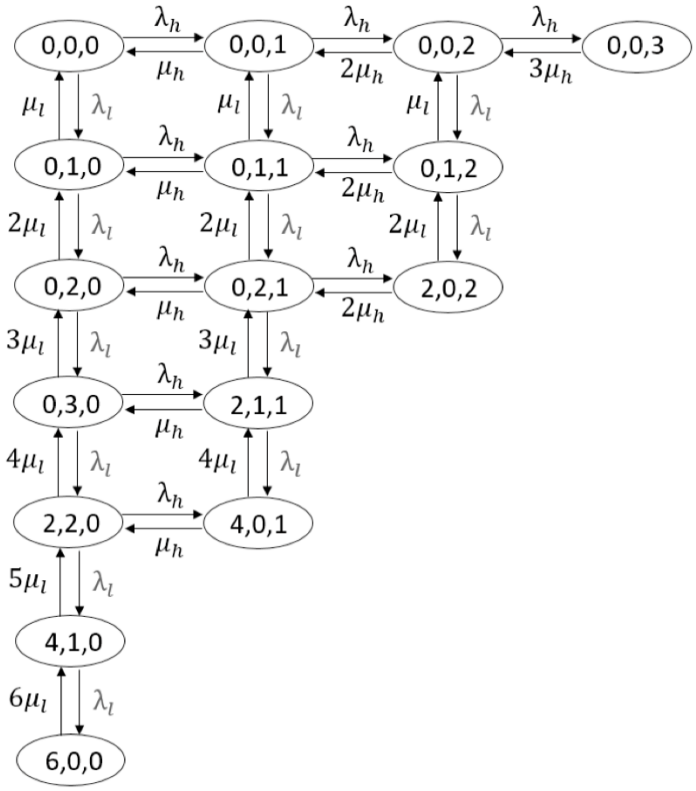
**Output:**

(a) 1-value(500; hw2-2a): **9.9472e-001**

(b) 1-value (500; hw2-2b): **9.9468e-001**

Problem #3

The Markov model is as follows:




**Source Code:**

bind
lambdah 3
muh 3
lambdal 4
mul 4
end

markov hw2-3
000 001 lambdah
001 002 lambdah
002 003 lambdah
010 011 lambdah
011 012 lambdah
020 021 lambdah
021 202 lambdah
030 211 lambdah
220 401 lambdah

```
000 010 lambdal
010 020 lambdal
020 030 lambdal
030 220 lambdal
220 410 lambdal
410 600 lambdal
001 011 lambdal
011 021 lambdal
021 211 lambdal
211 401 lambdal
002 012 lambdal
012 202 lambdal
001 000 muh
002 001 muh*2
003 002 muh*3
011 010 muh
012 011 muh*2
021 020 muh
202 021 muh*2
211 030 muh
401 220 muh
600 410 mul*6
410 220 mul*5
220 030 mul*4
030 020 mul*3
020 010 mul*2
010 000 mul
401 211 mul*4
211 021 mul*3
021 011 mul*2
011 001 mul
202 012 mul*2
012 002 mul
end

echo ------------------------------------------------
echo Q1: the rejection probability of high-priority clients
echo ------------------------------------------------
expr prob(hw2-3,600) + prob(hw2-3,410) + prob(hw2-3,401) + prob(hw2-3,211) +
prob(hw2-3,202) + prob(hw2-3 ,012) + prob (hw2-3,003)

echo ------------------------------------------------
echo Q2: the average number of high-priority clients in the system
expr prob(hw2-3,001) + prob(hw2-3,002)*2 + prob(hw2-3,003)*3 + prob(hw2-3,011) +
prob(hw2-3,012)*2 + prob(hw2-3,021) + prob(hw2-3,202)*2 + prob(hw2-3,211) +
prob(hw2-3,401)
```

```
echo ------------------------------------------------
echo Q3: the throughput of high-priority clients
expr (prob(hw2-3,001) + prob(hw2-3,002)*2 + prob(hw2-3,003)*3 + prob(hw2-3,011) +
prob(hw2-3,012)*2 + prob(hw2-3,021) + prob(hw2-3,202)*2 + prob(hw2-3,211) +
prob(hw2-3,401) )*muh

echo ------------------------------------------------
echo Q4: the throughput of low-priority, low QoS clients
expr prob(hw2-3,202)*2*mul + prob(hw2-3,211)*2*mul + prob(hw2-3,220)*2*mul +
prob(hw2-3,401)*4*mul + prob(hw2-3,410)*4*mul + prob(hw2-3,600)*6*mul
end
```

**Output:**

```
* ------------------------------------------------
* Q1: the rejection probability of high-priority clients
* ------------------------------------------------
 prob(hw2-3,600) + prob(hw2-3,410) + prob(hw2-3,401) + prob(hw2-3,211) + prob(hw2-3,202)
+ prob(hw2-3 ,012) + prob (hw2-3,003):   1.6582e-001

* ------------------------------------------------
* Q2: the average number of high-priority clients in the system
------------------------------------------

 prob(hw2-3,001) + prob(hw2-3,002)*2 + prob(hw2-3,003)*3 + prob(hw2-3,011) + prob(hw2-
3,012)*2 + prob(hw2-3,021) + prob(hw2-3,202)*2 + prob(hw2-3,211) + prob(hw2-3,401):
8.3418e-001

* ------------------------------------------------
* Q3: the throughput of high-priority clients
------------------------------------------

 (prob(hw2-3,001) + prob(hw2-3,002)*2 + prob(hw2-3,003)*3 + prob(hw2-3,011) + prob(hw2-
3,012)*2 + prob(hw2-3,021) + prob(hw2-3,202)*2 + prob(hw2-3,211) + prob(hw2-3,401) )*muh:
2.5025e+000

* ------------------------------------------------
* Q4: the throughput of low-priority, low QoS clients
------------------------------------------

 prob(hw2-3,202)*2*mul + prob(hw2-3,211)*2*mul + prob(hw2-3,220)*2*mul + prob(hw2-
3,401)*4*mul + prob(hw2-3,410)*4*mul + prob(hw2-3,600)*6*mul:   6.5760e-001
```