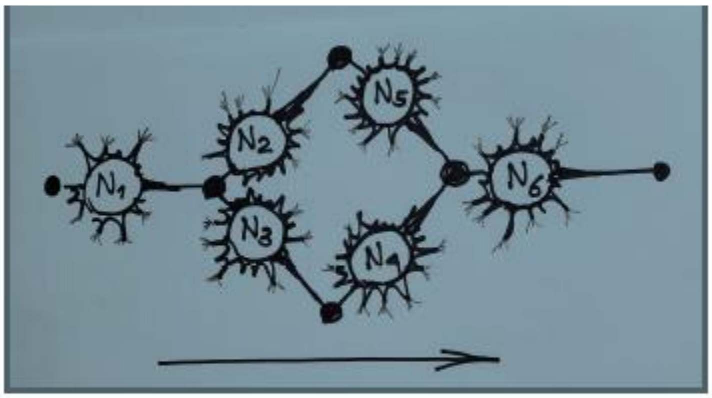
Problem 1:

Bayes Net:

Denote note N being fired: N =1, N = 0 otherwise



Given N1 is stimulated:

- P(N1 = 1) = 0.9

Probability of a node is being fired given its parent:

- P(Ni = 1|Ni-1 = 1) = 0.9
- P(Ni =1|Ni-1 =0) = 0.05
- (i,i-1) in (2,5), (3,4)

Probability of node N6 is being fired given its parents:

- P(N6 = 1|N5 = 1, N4 = 1) = 0.9
- P(N6 = 1|N5 = 0, N4 = 1) = 0.9
- P(N6 = 1|N5 = 1, N4 = 0) = 0.9
- P(N6 = 1|N5 = 0, N4 = 0) = 0.05

Using variable elimination:

a) Probability that N6 will be fired is 0.900453
b) Probability that N6 will be fire is N4 did not fire is 0.9047627
c) Probability that N5 received stimulus if N6 did not fire is 0.904794

Problem 2:

$$P(X_i|\lambda) \sim Ga(r, \lambda)$$

$$\pi(\lambda) \sim Ga(\alpha, \beta)$$

$$f(\lambda, X1, X2, \ldots X23) = \frac{\prod Xi^{r-1}}{\Gamma(r)^{23}} \frac{\beta^{\alpha}}{\Gamma(\alpha)} \lambda^{23r+\alpha-1} e^{-\lambda(\beta + \Sigma x_i)}$$

Therefore,

$\lambda|(X1, X2, \ldots X23) \sim \mathrm{Ga}(\alpha + \mathrm{n}r, \beta + \Sigma x_i)$

Bayesian estimator of λ: $\lambda = \frac{(\alpha + \mathrm{n}r)}{\beta + \Sigma x_i}$ = 0.5551

The 95% equitailed credible set for λ is [0.44908894, 0.67212117]

Posterior (H: $\lambda < 0.5\ |X) = 0.16688$

Problem 3: Gibbs Sampler and Mating Call

Denote:

$$Y_1 = (y_{1,i})|i = 1..43$$

$$Y_2 = (y_{2,i})|i = 1..2$$

For Y_1 we have:

$$\pi(\mu_1, \tau_1, Y_1) = f(Y1\,|\mu_1, \tau_1)\pi(\tau_1)\pi(\mu_1)$$
$$\propto \exp\{-0.5\ (19.5283\,\tau_1 - (55.9 + 1.2)\tau_1\ \mu_1 + (43\tau_1 + 1){\mu_1}^2 + 0.36)\}$$

Therefore,

$$\mu_1|Y1, \tau_1 \sim N(\frac{27.98\ \tau_1 + 0.6}{43\tau_1 + 1}, \frac{1}{43\tau_1 + 1})$$

Similarly,

$$\tau_1|\mu_1, Y1 \sim Ga(41.5, 0.5 + 0.5\sum_{i=1}^{43}(\overline{Y1} - \mu_1)\text{^}2)$$

$$\mu_2|Y2, \tau_2 \sim N(\frac{6.48\ \tau_2 + 0.6}{12\tau_2 + 1}, \frac{1}{12\tau_1 + 1})$$

$$\tau_2|\mu_2, Y2 \sim Ga\left(26, 0.5 + 0.5\sum_{i=1}^{12}(\overline{Y2} - \mu_2)\text{^}2\right)$$

a) After 11000 simulations, 95% credible set for $\mu_1 - \mu_2\ is\ \ [0.09609863, 0.108000456]$

This set does not contain 0, therefore we can reject the null hypothesis

H0: $\mu_1 = \mu_2$

Based on this analysis, the length of the call is a discriminatory characteristic since Hyla chrysoscelis exhibit longer length of the call than Hyla versicolor.

Code for Problem 1:

```
import sys


from numpy import zeros, float32
#  pgmpy
import pgmpy
from pgmpy.models import BayesianModel
from pgmpy.factors.discrete import TabularCPD
from pgmpy.inference import VariableElimination
```

```
def make_power_plant_net():
    """Create a Bayes Net representation of the above power plant problem. """

    BayesNet = BayesianModel()
    nodes = ['N1', 'N2', 'N3', 'N4', 'N5', 'N6']
    BayesNet.add_nodes_from(nodes)

    edges = [('N1', 'N2'), ('N1', 'N3'), ('N2', 'N5'), ('N3', 'N4'), ('N5', 'N6'), ('N4', 'N6')]
    BayesNet.add_edges_from(edges)

    return BayesNet
```

```
def set_probability(bayes_net):
    """Set probability distribution for each node in the power plant system."""

    # TODO: set the probability distribution for each node
    cpd_N1 = TabularCPD('N1', 2, values =[[0.1 , 0.9]])
    cpd_N2 = TabularCPD('N2', 2, values =[[0.05 , 0.1],\
                                          [0.95 , 0.9]], evidence =['N1'], evidence_card =[2] )
    cpd_N3 = TabularCPD('N3', 2, values =[[0.05 , 0.1],\
                                          [0.95 , 0.9]], evidence =['N1'], evidence_card =[2] )
    cpd_N4 = TabularCPD('N4', 2, values =[[0.05 , 0.1],\
                                          [0.95 , 0.9]], evidence =['N3'], evidence_card =[2] )
    cpd_N5 = TabularCPD('N5', 2, values =[[0.05 , 0.1],\
                                          [0.95 , 0.9]], evidence =['N2'], evidence_card =[2] )
    cpd_N6 = TabularCPD('N6', 2, values =[[0.05 , 0.1, 0.1, 0.1],\
                                          [0.95 , 0.9, 0.9, 0.9]], evidence =['N4', 'N5'], evidence_card =[2, 2]
    bayes_net.add_cpds(cpd_N1,cpd_N2,cpd_N3,cpd_N4,cpd_N5,cpd_N6 )

    return bayes_net
```

```
bayes_net = make_power_plant_net()
```

```
bayes_net = set_probability(bayes_net)
```

```python
def get_N6_prob(bayes_net):
    """Calculate the marginal"""

    solver = VariableElimination(bayes_net)
    conditional_prob = solver.query(variables=['N6'],  joint=False)
    prob = conditional_prob['N6'].values
    N6_prob = prob[1]
    return N6_prob
```

```python
get_N6_prob(bayes_net)
```

```
Finding Elimination Order: : 100%|██████████| 5/5 [00:00<00:00, 733.37it/s]
Eliminating: N2: 100%|██████████| 5/5 [00:00<00:00, 165.18it/s]
```

```
0.9004536562500001
```

```python
"""probability that N6 will fire is 0.900453"""
```

```
'probability that N6 will fire is 0.900453'
```

```python
def get_N6_prob_N4(bayes_net):

    solver = VariableElimination(bayes_net)
    conditional_prob = solver.query(variables=['N6'],evidence={'N4':0}, joint=False)
    N6_N4_prob = conditional_prob['N6'].values
    return N6_N4_prob[1]
```

```python
get_N6_prob_N4(bayes_net)
```

```
Finding Elimination Order: : 100%|██████████| 4/4 [00:00<00:00, 1023.31it/s]
Eliminating: N2: 100%|██████████| 4/4 [00:00<00:00, 273.06it/s]
```

```
0.9047627952755906
```

```python
def get_N5_prob(bayes_net):

    # TODO: finish this function
    solver = VariableElimination(bayes_net)
    conditional_prob = solver.query(variables=['N2'],evidence={'N6':0}, joint=False)
    N5_prob = conditional_prob['N2'].values
    return N5_prob[1]
```

```python
get_N5_prob(bayes_net)
```

```
Finding Elimination Order: : 100%|██████████| 4/4 [00:00<00:00, 2046.00it/s]
Eliminating: N1: 100%|██████████| 4/4 [00:00<00:00, 273.11it/s]
```

```
0.9047940296652031
```

Code for Problem 2:

```python
from scipy.stats import gamma
import numpy as np
```

```python
#gamma function with shape factor =1 and rate hyperparameter = 95
f = gamma(95)
```

```python
#calculate 95% equitailed credible set
x = f.interval(0.95)
```

```python
#calculate actual lambda
x = np.array(x) /171.148
```

```python
print("95% Equitailed Credible set for λ", x)
```

```
95% Equitailed Credible set for λ [0.44908894 0.67212117]
```

```python
#calculationg posterior of hypothesis
posterior = f.cdf(0.5*171.148)
```

```python
posterior
```

```
0.1668814437103798
```

Code for Problem 3

In [23]:
```
import numpy as np
import matplotlib.pyplot as plt
import numpy.random as rand
```

In [24]:
```
rand.seed(0)
def m1(mu_1):
    f = 42*0.18**2 + 43*(0.65-mu_1)**2
    return f
def m2(mu_2):
    f = 11*0.14**2 +12*(0.54-mu_2)**2
    return f
```

In [25]:
```
def normal_1(tau):
    mu = (27.98*tau+0.6)/(43*tau+1)
    sigma = 1/(43*tau +1)
    f = rand.normal(mu, sigma, 1)
    return f
def normal_2(tau):
    mu=(6.48*tau+0.6)/(12*tau+1)
    sigma = 1/(12*tau +1)
    f= rand.normal(mu,sigma,1)
    return f
def gamma_1(mu):
    alpha = 41.5
    beta = float(0.5+0.5*m1(mu))
    f = rand.gamma(alpha, beta)
    return f
def gamma_2(mu):
    alpha = 26
    beta = float(0.5 + 0.5*m2(mu))
    f = rand.gamma(alpha, beta)
    return f
```

In [26]:
```
## Initialization
mu_1 =0
mu_2 = 0
tau_1 =10
tau_2 = 10
for i in range(0,1000):
    mu_1 = normal_1(tau_1)
    mu_2 = normal_2(tau_2)
    tau_1 = gamma_1(mu_1)
    tau_2 = gamma_2(mu_2)
```

In [27]:
```
result_1 = np.array([])
result_2 = np.array([])
result_diff = np.array([])
```

```
0]: count = 0
for i in range(10000):
    mu_1 = normal_1(tau_1)
    mu_2 = normal_2(tau_2)
    tau_1 = gamma_1(mu_1)
    tau_2 = gamma_2(mu_2)
    if mu_1 >mu_2:
        count +=1
        result_1 = np.append(result_1, np.array(mu_1), 0)
        result_2 = np.append(result_2, np.array(mu_2), 0)
        result_diff = np.append(result_diff, np.array(mu_1-mu_2), 0)

print(count/10000)
estimate_1 = np.mean(result_1)
estimate_2 = np.mean(result_2)

result_sort = np.sort(result_diff)
print("After 11000 simulations, Bayesian Estimator for µ_1 is", estimate_1)
print("After 11000 simulations, Bayesian Estimator for µ_2 is", estimate_2)
print("95% credible set for µ_1 - µ_2 is" ,[result_sort[249], result_sort[9749]])
```

```
1.0
After 11000 simulations, Bayesian Estimator for µ_1 is 0.650674563363189
After 11000 simulations, Bayesian Estimator for µ_2 is 0.540355006109903
95% credible set for µ_1 - µ_2 is [0.09609863038507038, 0.10800045665842561]
```