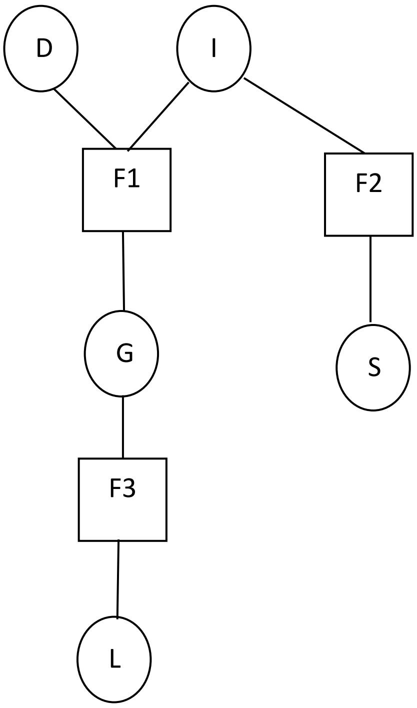
GMDL, HW#3

208727164 – אורן יקואל
336132014 - יונתן לויטס
318512910 - גל אלון



With the attached code, we calculated the marginals and those are our results:

$p(i)$ = [0.7 0.3]
$p(d)$ = [0.6 0.4]
$p(g)$ = [0.362 0.2884 0.3496]
$p(s)$ = [0.725 0.275]
$p(l)$ = [0.497664 0.502336]

gmdl-hw3

May 29, 2023

[1]:

```python
import numpy as np
```

[4]:

```python
pd = np.asarray([0.6, 0.4])

pi = np.asarray([0.7, 0.3])

pg = np.asarray([[0.3, 0.4, 0.3],
                 [0.05, 0.25, 0.7],
                 [0.9, 0.08, 0.02],
                 [0.5, 0.3, 0.2]])

ps = np.asarray([[0.95, 0.05],
                 [0.2, 0.8]])

pl = np.asarray([[0.1, 0.9],
                 [0.4, 0.6],
                 [0.99, 0.01]])
```

[13]:

```python
# f1 is a 3d array with shape (3, 2, 2) a function of g, d, i respectively, axis 0 = g, axis 1 = d, axis 2 = i
f1 = np.zeros((3, 2, 2))

for g in range(3):
    for d in range(2):
        for i in range(2):
            f1[g, d, i] = pi[i] * pd[d] * pg[2 * i + d, g]

#f2 is a 2d array with shape (2, 2) a function of i, s respectively, axis 0 = i, axis 1 = s
f2 = np.copy(ps)

#f3 is a 2d array with shape (3, 2) a function of g, l respectively, axis 0 = g, axis 1 = l
f3 = np.copy(pl)
```

[15]:

```python
# trivial factors
m_d_f1 = np.ones(2)
```

```
m_s_f2 = np.ones(2)
m_l_f3 = np.ones(3)
```

[30]:

```
# m_f3_g = sum over l of f3 * m_l_f3
m_f3_g = np.sum(f3, axis=1) * m_l_f3

# m_f2_i = sum over s of f2 * m_s_f2
m_f2_i = np.sum(f2, axis=1) * m_s_f2

#m_g_f1 = multiply over f/f3 (neighboring factors of g) m_f_g
m_g_f1 = np.copy(m_f3_g)

# m_f1_i = sum over g,d of f1 * m_g_f1 * m_d_f1
m_f1_i = np.zeros(2)

for g in range(3):
    for d in range(2):
        m_f1_i += f1[g, d, :] * m_g_f1[g] * m_d_f1[d]


m_i_f1 = np.copy(m_f2_i)

# m_f1_d = sum over g,i of f1 * m_g_f1 * m_i_f1
m_f1_d = np.zeros(2)

for g in range(3):
    for i in range(2):
        m_f1_d += f1[g, :, i] * m_g_f1[g] * m_i_f1[i]


# m_f1_g = sum over d,i of f1 * m_d_f1 * m_i_f1

m_f1_g = np.zeros(3)

for d in range(2):
    for i in range(2):
        m_f1_g += f1[:, d, i] * m_d_f1[d] * m_i_f1[i]

m_g_f3 = np.copy(m_f1_g)

#m_f3_l = sum over g of f3 * m_g_f3

m_f3_l = np.zeros(2)

for g in range(3):
    m_f3_l += f3[g, :] * m_g_f3[g]
```

```
m_i_f2 = np.copy(m_f1_i)

# m_f2_s = sum over i of f2 * m_i_f2

m_f2_s = np.zeros(2)

for i in range(2):
    m_f2_s += f2[i, :] * m_i_f2[i]
```

[21]:

```
p_i = m_f1_i * m_f2_i
print(p_i)
```

```
[0.7 0.3]
```

[24]:

```
p_d = np.copy(m_f1_d)
print(p_d)
```

```
[0.6 0.4]
```

[26]:

```
p_g = m_f1_g * m_f3_g
print(p_g)
```

```
[0.362  0.2884 0.3496]
```

[28]:

```
p_l = np.copy(m_f3_l)
print(p_l)
```

```
[0.497664 0.502336]
```

[31]:

```
p_s = np.copy(m_f2_s)
print(p_s)
```

```
[0.725 0.275]
```