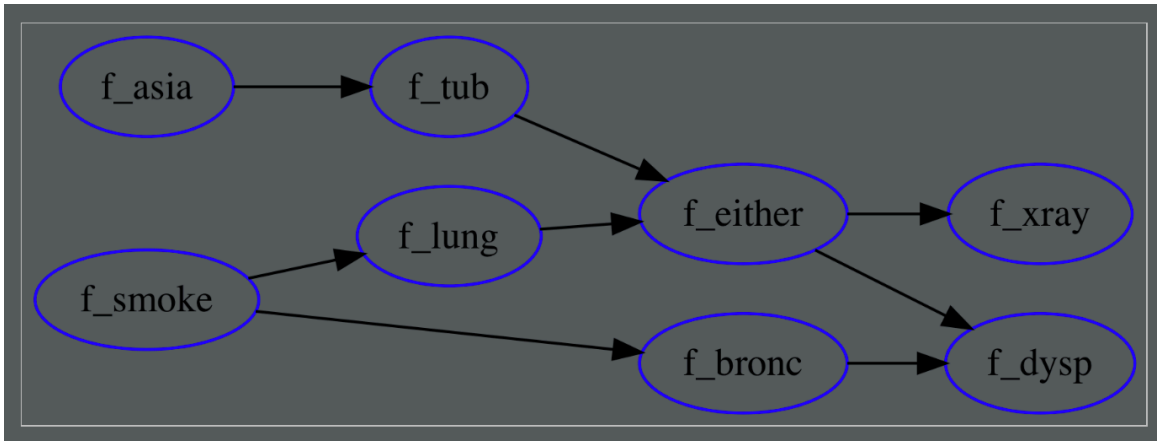
Problem Set 2

1 Problem 1

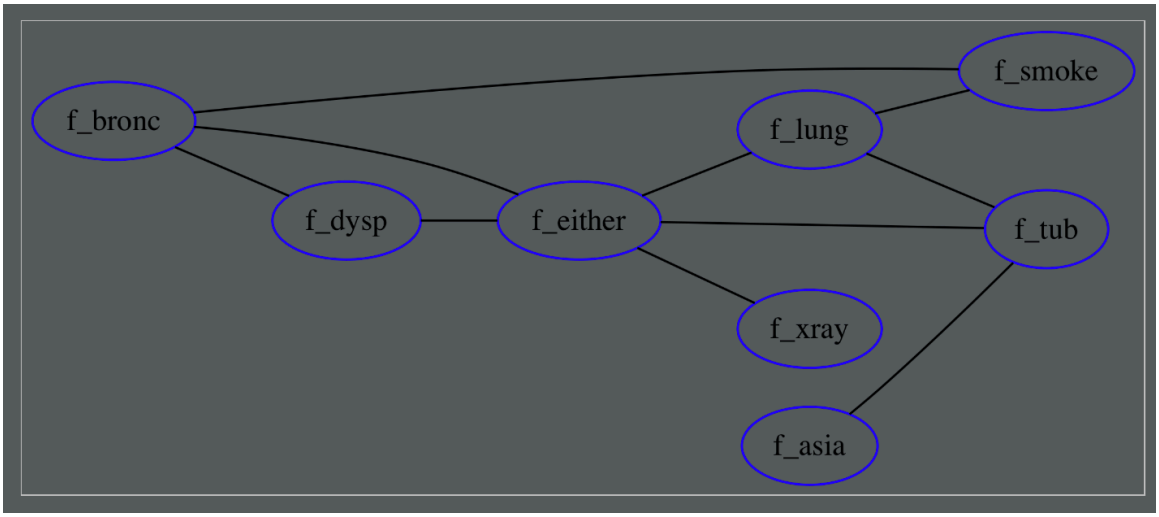
1.1 Load data and create BNN

```
import bif_parser
import prettytable
imp  Unused import statement more... (⌘F1)
from IPython.core.display import Image
from bayesian.bbn import *
name = 'asia'
module_name = bif_parser.parse(name)
module = __import__(module_name)
bg = module.create_bbn()


def show_graphgiz_image(graphviz_data):
    graph = pydot.graph_from_dot_data(graphviz_data)
    graph[0].write_png('temp.png')
    return 'temp.png'

sf = bg.get_graphviz_source()
Image(filename=show_graphgiz_image(sf))
```



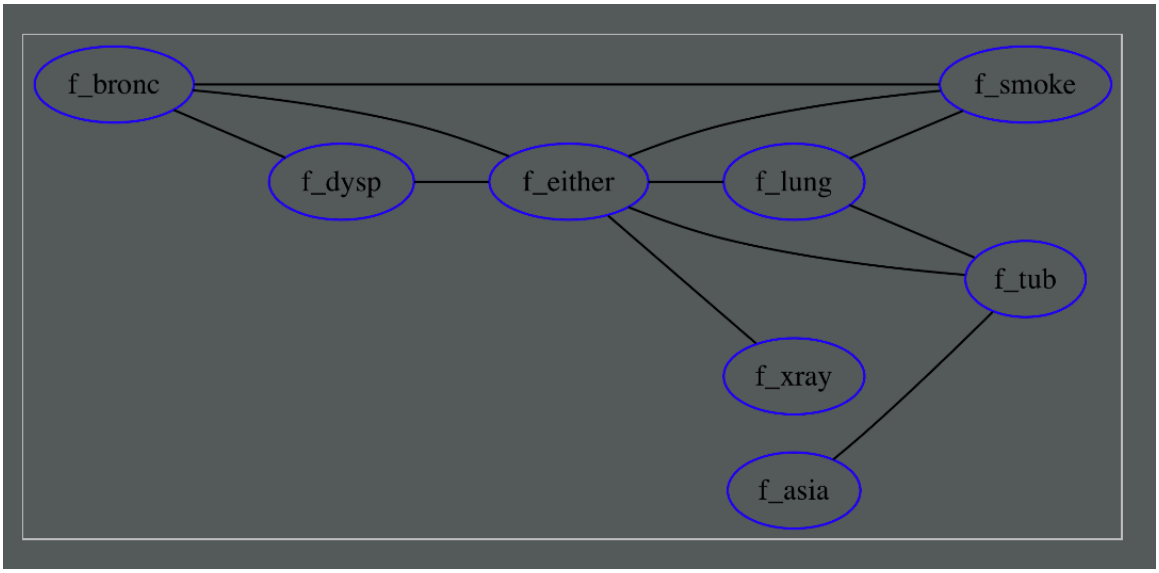
1.2 Moralization

```
gu=make_undirected_copy(bg)
m1=make_moralized_copy(gu,bg)
s2=m1.get_graphviz_source()
Image(filename=show_graphgiz_image(s2))
```



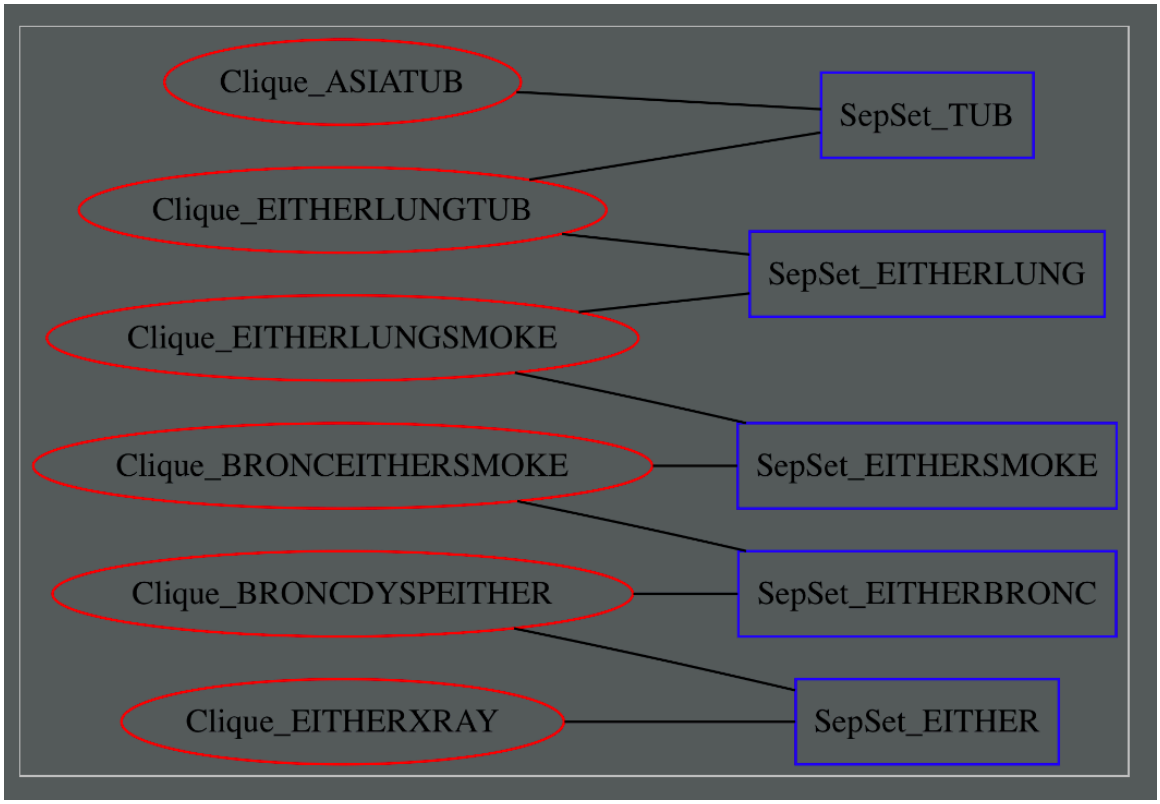
1.3 Triangulation

```
cliques, elimination_ordering = triangulate(m1, priority_func)
s2 = m1.get_graphviz_source()
Image(filename=show_graphgiz_image(s2))
```



1.4 Build the join tree

```
jt = bg.build_join_tree()
sf = jt.get_graphviz_source()
Image(filename=show_graphgiz_image(sf))
```



1.5 Propagate

```
assignments = jt.assign_clusters(bg)
jt.initialize_potentials(assignments, bg)
jt.propagate()
```

1.6 Running intersection property

For each pair of clusters B and C that contain I, each cluster on the unique path between B and C also contain i.

I create cliques according to the preceding graph and create the sepsets which are the intersection points between every pair of cliques. It can be seen the sepsets contain the intersection points of variables from the cliques that they connect with.

Prove:

Assume my junction tree does not satisfy the Running intersection property. It means that the message of a variable can be spread through more than one channels, so the message should be considered more than once. When I call the function propagate(), this will run message passing between all the variables, but cannot get consistent beliefs in all its clusters.

Just display one case: choose two different cliques Clique_BRONCDYSPEITHER and Clique_BRONCEITHERSMOKE to compute the marginal values.

In [7]:
```
bronc_clust=[i for i in jt.clique_nodes for v in i.variable_names if
pot=bronc_clust[0].potential_tt
#a function to return the sum for a specific assignment, such as'bron
sum_assignments=lambda imap,tup:sum([v for k,v in imap.iteritems() fo
#get the sum for bronc=yes and bronc=no
yes,no=[sum_assignments(pot,('bronc',i)) for i in ['yes','no']]
print 'bronc: yes ', yes/float(yes+no)," no ", no/float(yes+no)
```

```
bronc: yes  0.45  no  0.55
```

In [8]:
```
pot2=bronc_clust[1].potential_tt
yes,no=[sum_assignments(pot2,('bronc',i)) for i in ['yes','no']]
print 'bronc: yes ', yes/float(yes+no)," no ", no/float(yes+no)
```

```
bronc: yes  0.45  no  0.55
```

We can obtain the same values of the marginal of bronc, indicating that the beliefs for all the variables are consistent across all clusters and sepsets. So my junction tree satisfies the Running intersection property.

1.7 Find the joint probability

"tub=yes, lung=yes, bronc=yes", given evidence that "asia=yes, xray=yes".

```
library(gRain)
yn <- c("yes","no")
a <- cptable(~asia, values=c(1,99), levels=yn)
t.a <- cptable(~tub | asia, values=c(5,95,1,99), levels=yn)
s <- cptable(~smoke, values=c(5,5), levels=yn)
l.s <- cptable(~lung | smoke, values=c(1,9,1,99), levels=yn)
b.s <- cptable(~bronc | smoke, values=c(6,4,3,7), levels=yn)
e.lt <- cptable(~either | lung:tub,values=c(1,0,1,0,1,0,0,1),levels=yn)
x.e <- cptable(~xray | either, values=c(98,2,5,95), levels=yn)
d.be <- cptable(~dysp | bronc:either, values=c(9,1,7,3,8,2,1,9),levels=yn)
cpt.list <- compileCPT(list(a, t.a, s, l.s, b.s, e.lt, x.e, d.be))
cpt.list
bnet <- grain(cpt.list)
bnet <- compile(bnet)
bnet
jointree <-triangulate(moralize(bnet$dag))
par(mfrow=c(1,2)); plot(c)
jointree.ev<-setEvidence(jointree, nodes = c("asia","xray"),states = c("yes","yes"))
x<-querygrain(jointree.ev, nodes=c("tub","lung","bronc"),type="joint")
ftable(x, row.vars=1)
```

Set evidence asia = 'yes', xray = 'yes'

lung	yes		no	
bronc	yes	no	yes	no
tub				
yes	0.010638041	0.007936316	0.141333977	0.177807261
no	0.202122784	0.150790013	0.137007426	0.172364181

P(tub = yes, lung=yes, bronc=yes | asia=yes, xray=yes) = 0.010638041

2 Problem 2:

2.1 Describe how the different terms on the right hand side of "p(V) = p(a)p(t | a)p(s)p(l | s)p(b | s)p(e | t, l)p(d | e, b)p(x | e)" are distributed among the different juction tree clusters.

Clique_ASIATUB p(a)p(t | a)=Ø(a,t)

Clique_EITHERLUNGTUB p(e|t,l) =Ø(e,t,l)

Clique_EITHERXRAY p(x|e) =Ø(x,e)

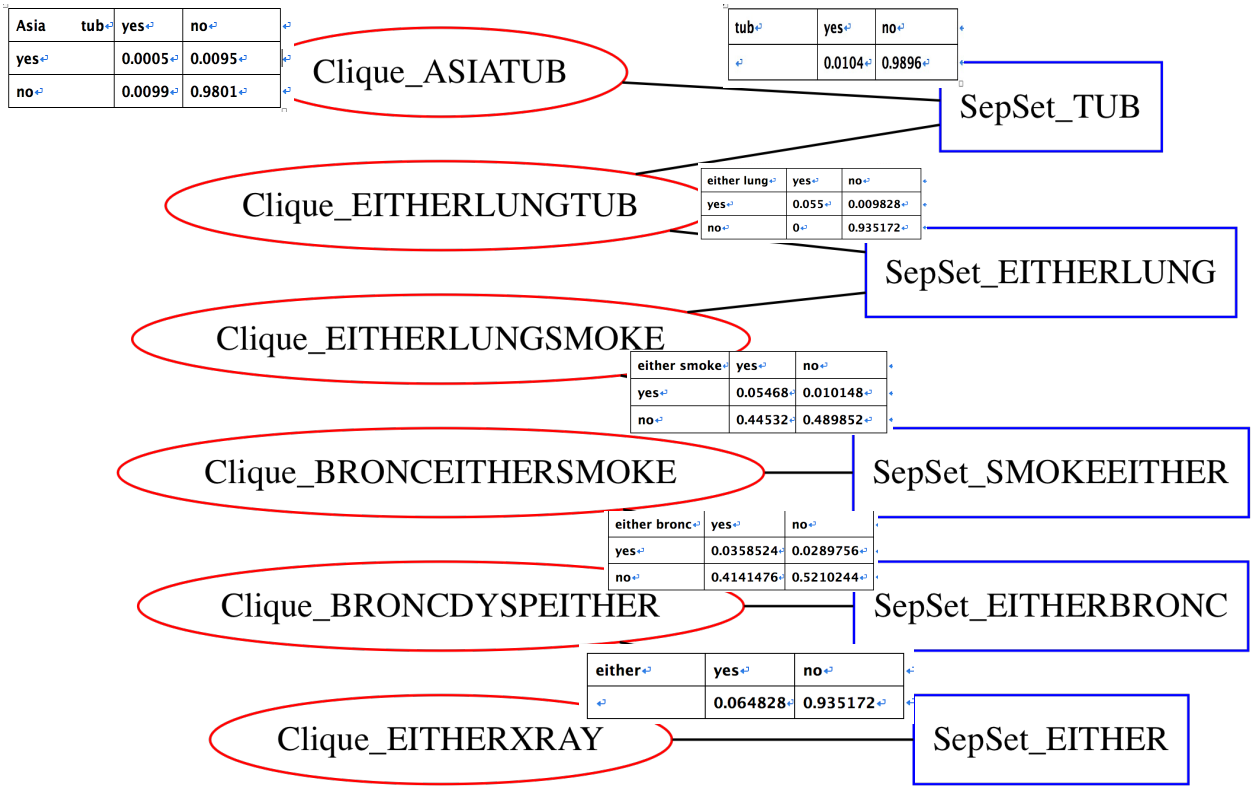
Clique_EITHERLUNGSMOKE $p(l|s) = \sum_e P(e,l|s) = \sum_e Ø(e,l,s)$

Clique_BRONCEITHERSMOKE $p(b|s) = \sum_e P(e,b|s) = \sum_e Ø(e,b,s)$

Clique_BRONCDYSPEITHER p(d|e,b) =Ø(d,e,b)

2.2 Write out the messages using these terms and verify that the message passing algorithm indeed gives the cluster marginals.

Select Clique_ASIATUB as starting clique.



pass the message from starting clique.

There are three actors in a message pass: the source clique, the intervening sepset, and the destination clique. Each of these has a set of potentials (very similar to a CPD along with the associated probabilities, except that it need not be a valid probability distribution).

For example, a message from φx (bronc,either,smoke)to φy (either,lung,smoke) would pass the φr (either, smoke) sepset, where the variables are in the scope of each clique or sepset.

Unmark all cliques, collect evidence from starting clique. Pass message from sender to receiver.

The message will modify the potentials at φr and φ y, that is, the sepset potential and the destination cluster potential. The rst assignment to the sepset R are the potentials in the source with the variables not in the sepset marginalized out, as shown in the following formula:

$$\varphi \mathrm{R} = \sum_{X/R} \varphi \mathrm{x}$$

P(tub) = $\sum_{asia} P(asia, tub)$

P(either,lung) = $\sum_{tub} P(either, lung, tub)$

P(smoke,either) = $\sum_{bronc} P(bronc, either, smoke)$

P(either,bronc) = $\sum_{smoke} P(bronc, either, smoke)$

P(either) = $\sum_{bronc} \sum_{dysp} P(bronc, dysp, either)$

Distribute evidence. Update potentials

The second assignment to the cluster Y is as follows:

$$\varphi \mathrm{Y} = \varphi \mathrm{Y} \frac{\varphi \mathrm{R}}{\varphi_R^{old}}$$

Reference:

[1]Building Probabilistic Graphical Models with Python (https://ublearns.buffalo.edu/bbcswebdav/pid-4499403-dt-content-rid-18203418_1/courses/2181_20888_PsC/Building%20Probabilistic%20Graphical%20Models%20with%20Python%20%5BKarkera%202014-05-25%5D.pdf)

[2]http://people.math.aau.dk/~sorenh/misc/2014-useR-GMBN/bayesnet-slides.pdf