1. **Bayesian Inference**

Consider two factors that influence whether a student passes a given test: a) being smart, and b) studying. Suppose 30% of students believe they are intrinsically smart. But since students do not know a priori whether they are smart enough to pass a test, suppose 40% of will study for it anyway. (assume Smart and Study are independent). The causal relationship of these variables on the probability of actually passing the test can be expressed in a conditional probability table (CPT) as follows:

P(pass|Smart,Study)	¬smart	smart
¬study	0.2	0.7
study	0.6	0.95

prior probabilities: P(smart)=0.3, P(study)=0.4

   a. **Write out the equation for calculating joint probabilities, P(Smart,Study,Pass).**

P(Smart, Study, Pass) = P(Pass | Smart, Study) x P(Smart) x P(Study)

b. **Calculate all the entries in the full joint probability table (JPT) [a 4x2 matrix, like Fig 12.3 in the textbook;**
    i. **[Note: names of variables are capitalized, lower-case indicates truth value, e.g. 'pass' means Pass=T, and '-pass' means Pass=F.]**

We need the following probabilities, calculated by the following equations:

- P(Smart, Study, Pass) = P(Pass | Smart, Study) x P(Smart) x P(Study)
    - 0.95 x 0.3 x 0.4 = 0.114
- P(Smart, Study, -Pass) = (1 - P(Pass | Smart, Study)) x P(Smart) x P(Study)
    - 0.05 x 0.3 x 0.4 = 0.006
- P(Smart, -Study, Pass) = P(Pass | Smart, -Study) x P(Smart) x P(-Study)
    - 0.7 x 0.3 x 0.6 = 0.126
- P(Smart, -Study, -Pass) = (1 - P(Pass | Smart, -Study)) x P(Smart) x P(-Study)
    - 0.3 x 0.3 x 0.6 = 0.054
- P(-Smart, Study, Pass) = P(Pass | -Smart, Study) x P(-Smart) x P(Study)
    - 0.6 x 0.7 x 0.4 = 0.168
- P(-Smart, Study, -Pass) = (1 - P(Pass | -Smart, Study)) x P(-Smart) x P(Study)
    - 0.4 x 0.7 x 0.4 = 0.112
- P(-Smart, -Study, Pass) = P(Pass | -Smart, -Study) x P(-Smart) x P(-Study)
    - 0.2 x 0.7 x 0.6 = 0.084
- P(-Smart, -Study, -Pass) = (1 - P(Pass | -Smart, -Study)) x P(-Smart) x P(-Study)
    - 0.8 x 0.7 x 0.6 = 0.336

	pass		**-pass**	
	study	**-study**	**study**	**-study**
smart	0.114	0.126	0.006	0.054
-smart	0.168	0.084	0.112	0.336

c. **From the JPT, compute the probability that a student is smart, given that they pass the test but did not study.**

P(Smart | Pass, -Study) = (0.126) / (0.126 + 0.084) = 0.6

d. **From the JPT, compute the probability that a student did not study, given that they are smart but did not pass the test.**

P(-Study | Smart, -Pass) = (0.054) / (0.054 + 0.006) = 0.9

e. **Compute the marginal probability that a student will pass the test given that they are smart.**

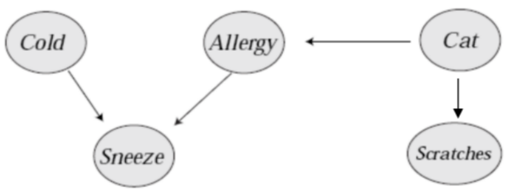
P(Pass | Smart) = (0.114 + 0.126) / (0.114 + 0.126 + 0.006 + 0.054) = 0.8

f. **Compute the marginal probability that a student will pass the test given that they study.**

P(Pass | Study) = (0.114 + 0.168) / (0.114 + 0.168 + 0.006 + 0.112) = 0.705

2. **Bayesian Networks**

Here is a probabilistic model that describes what it might mean when a person sneezes, e.g. depending on whether they have a cold, or whether a cat is present and they are allergic. Scratches on the furniture would be evidence that a cat had been present.



Cold = False	Cold = True
.95	.05

Cat Presence = False	Cat Presence = True
.98	.02

Cat Presence	Allergic Reaction = False	Allergic Reaction = True
False	.95	.05
True	.25	.75

Cold	Allergic Reaction	Sneeze = False	Sneeze = True
False	False	.99	.01
False	True	.3	.7
True	False	.2	.8
True	True	.1	.9

Cat Presence	Scratches = False	Scratches = True
False	.95	.05
True	.5	.5

a. **Using Equation 13.2 in the textbook (p. 415), write out the expression for the joint probability for any state (i.e. combination of truth values for the 5 variables in this problem).**
   i. **[Note: Use capital letters for names of variables, and lower-case to indicate truth value, e.g. 'cold' means Cold=T, and '-cold' means Cold=F.]**

P(Cold, Sneeze, Allergic, Scratches, Cat)
= P(Cold) x P(Sneeze | Cold, Allergy) x P(Allergy | Cat) x P(Scratches | Cat) x P(Cat)

b. **Use the equation above to calculate the joint probability that the person sneezes, but does not have a cold, has a cat, is allergic, and there are scratches on the furniture:**

P(-cold, sneeze, allergic, scratches, cat)
= P(-cold) x P(sneeze | -cold, allergy) x P(allergy | cat) x P(scratches | cat) x P(cat)
= 0.95 x 0.7 x 0.75 x 0.5 x 0.02
= 0.0049875

c. **Use normalization to calculate the conditional probability that a person has a cat, given that they sneeze and are allergic to cats, but do not have a cold, and there are scratches on the furniture.**

P(cat | -cold, sneeze, allergic, scratches)
= (P(cat, -cold, sneeze, allergic, scratches)) / (P(-cold, sneeze, allergic, scratches))

Where P(-cold, sneeze, allergic, scratches)
= P(cat, -cold, sneeze, allergic, scratches) + P(-cat, -cold, sneeze, allergic, scratches)

So P(cat | -cold, sneeze, allergic, scratches)
= α(P(cat, -cold, sneeze, allergic, scratches))
Where α
= 1 / (P(cat, -cold, sneeze, allergic, scratches) + P(-cat, -cold, sneeze, allergic, scratches))

So P(cat | -cold, sneeze, allergic, scratches)
= (P(cat, -cold, sneeze, allergic, scratches)) / (P(cat, -cold, sneeze, allergic, scratches) + P(-cat, -cold, sneeze, allergic, scratches))
= (Numerator) / (Denominator1 + Denominator2)

Numerator: P(cat, -cold, sneeze, allergic, scratches)
= P(cat) x P(-cold) x P(sneeze | allergic, -cold) x P(allergic | cat) x P(scratches | cat)
= 0.02 x 0.95 x 0.7 x 0.75 x 0.5
= 0.0049875

Denominator1: P(cat, -cold, sneeze, allergic, scratches)
= P(cat) x P(-cold) x P(sneeze | allergic, -cold) x P(allergic | cat) x P(scratches | cat)
= 0.02 x 0.95 x 0.7 x 0.75 x 0.5
= 0.0049875

Denominator2: P(-cat, -cold, sneeze, allergic, scratches)
= P(-cat) x P(-cold) x P(sneeze | allergic, -cold) x P(allergic | -cat) x P(scratches | -cat)
= 0.98 x 0.95 x 0.7 x 0.05 x 0.05
= 0.00162925

So (Numerator) / (Denominator1 + Denominator2) = (0.0049875) / (0.0049875 + 0.00162925)
= 0.75376884422

d. **Use Bayes' Rule to re-write the expression for P(cat|scratches). Look up the values for the numerator in the table above.**

P(cat | scratches)
= (P(scratches | cat) x P(cat)) / (P(scratches))
= (0.5 x 0.02) / (P(scratches))

e. **The denominator in the answer for (d) would require marginalization over how many joint probabilities? Write out the expressions for these (i.e. expand the denominator, but you don't have to calculate the actual values).**

To find P(scratches), we would have to sum out all joint probabilities where scratches is true - since there are 5 variables, and 4 of them have 2 possible states and 1 variable (scratches) has 1 state (true), there are a total of ((2^4) * 1) = 16 joint probabilities we would have to sum out - each probability is listed below. In order to calculate the actual values, we could expand each probability out like we did in parts a and b.

1. P(cat, cold, sneeze, allergic, scratches)
2. P(cat, cold, sneeze, -allergic, scratches)
3. P(cat, cold, -sneeze, allergic, scratches)
4. P(cat, cold, -sneeze, -allergic, scratches)
5. P(cat, -cold, sneeze, allergic, scratches)
6. P(cat, -cold, sneeze, -allergic, scratches)
7. P(cat, -cold, -sneeze, allergic, scratches)
8. P(cat, -cold, -sneeze, -allergic, scratches)
9. P(-cat, cold, sneeze, allergic, scratches)
10. P(-cat, cold, sneeze, -allergic, scratches)
11. P(-cat, cold, -sneeze, allergic, scratches)
12. P(-cat, cold, -sneeze, -allergic, scratches)
13. P(-cat, -cold, sneeze, allergic, scratches)
14. P(-cat, -cold, sneeze, -allergic, scratches)
15. P(-cat, -cold, -sneeze, allergic, scratches)
16. P(-cat, -cold, -sneeze, -allergic, scratches)

3. **PDDL and Situation Calculus**

To start a car, you have to be at the car and have the key, and the car has to have a charged battery and the tank has to have gas. Afterwards, the car will be running, and you will still be at the car and have the key after starting the engine.

a. **Write a PDDL operator to describe this action.**
   i. **(note: you can express this ego-centrically – you don't have to refer explicitly to the person starting the car; but the operator should take the car being started as an argument)**

```
(:action start-car
   :parameters (?car - Car)
   :precondition (and
                  (at ?car)
                  (has-key ?car)
                  (charged-battery ?car)
                  (has-gas ?car))
                 (not (car-running ?car))
   :effect (and
            (car-running ?car)
            (at ?car)
            (has-key ?car)))
```

b. **Describe the same operator using Situation Calculus (remember to add a situation argument to your predicates).**

Predicates:

- At(car, s) - The car is at the location in situation s.
- HasKey(car, s) - You have the key to the car in situation s.
- ChargedBattery(car, s) - The car has a charged battery in situation s.
- HasGas(car, s) - The car has gas in the tank in situation s.
- not CarRunning(car, s) - The car is running in situation s.

Action: StartCar(car, s)

Effect axioms:

- CarRunning(car, do(StartCar(car), s)) is true if:
- At(car, s) is true,
- HasKey(car, s) is true,
- ChargedBattery(car, s) is true, and
- HasGas(car, s) is true.
- At(car, do(StartCar(car), s)) is true if At(car, s) is true.
- HasKey(car, do(StartCar(car), s)) is true if HasKey(car, s) is true.

c. **Add a Frame Axiom that says that starting this car will not change whether any other car is out of gas (tank empty).**

Note: the predicate **'**HasGas(otherCar, s)' indicates that another car, 'otherCar', has gas in its tank in situation 's'.

Frame Axiom for the action StartCar:

- For any car 'car' and any other car 'otherCar' (where 'otherCar' is different from 'car'), and any situation 's', if 'HasGas(otherCar, s)' is true, then 'HasGas(otherCar, do(StartCar(car), s))' is also true.

$\forall$ car, otherCar, s. ((otherCar $\neq$ car) $\land$ HasGas(otherCar, s)) $\rightarrow$ HasGas(otherCar, do(StartCar(car), s))