1. Bayesian Inference

a) P(smart) = 0.3, P(study|smart) = 0.4, P(pass|smart,study) = 0.95
P(smart, study,pass) = P(smart) * P(studying|smart) * P(pass|smart,study) = 0.3 * 0.4 * 0.95 = *0.114*

b) P(-smart,-study,-pass) = (1-0.3) * (1-0.4) * (1-0.95) = 0.021
P(-smart,study,-pass) = (1-0.3) * (0.4) * (1- 0.95) = 0.014
P(smart,-study,-pass) = (0.3) * (1-0.4) * (1-0.95) = 0.009
P(smart,study,-pass) = 0.3 * 0.4 * (1-0.95) = 0.006
–
P(smart,study,pass) = 0.3 * 0.4 * 0.95 = 0.114
P(smart,-study,pass) = 0.3 * (1-0.4) * 0.95 = 0.171
P(-smart,study,pass) = (1-0.3) * 0.4 * 0.95 = 0.266
P(-smart,-study,pass) = (1-0.3) * (1-0.4) * 0.95 = 0.399

	smart		-smart	
	pass	-pass	pass	-pass
study	0.114	0.006	0.266	0.014
-study	0.171	0.009	0.399	0.021

c) P(smart|pass,study) = (P(pass|smart,-study) * P(smart|-study)) / P(pass|-study)

*P(pass|-study) = P(pass|smart,-study)*P(smart) + P(pass|-smart,-study)*P(-smart) = 0.7*0.3 + 0.2*0.7 = 0.35*

= (0.7 * 0.3) / 0.35 = **0.6, or 60% chance** that a student is smart given that they pass but did not study.

d) P(-study|smart,-pass) = ((P(-pass|smart,-study)*P(-study,smart)) / P(-pass|smart)

*P(-pass|smart) = P(-pass|smart,-study)*P(-study) + P(-pass|smart,study)*P(study) = 0.3*0.6 + 0.05*0.4 = 0.2*

= ((1-0.7)*(1-0.6)) / 0.2 = **0.6, or 60% chance** that a student did not study, given that they are smart but did not pass

e) P(pass|smart) = P(pass ^ smart) / (P(smart)) = (0.114 + 0.171) / (0.006 + 0.009 + 0.114 + 0.171) = **0.95**

f) P(pass|study) = P(pass ^ study) / P(study) = (0.114 + 0.266) / (0.4) = **0.95**

2. Bayesian Networks

a) P(cold,sneeze,allergic,scratches,cat) = P(cold) * P(sneeze|cold, allergic) * P(allergic|cat) * P(scratches|cat) * P(cat)

b) P(-cold,sneeze,allergic,scratches,cat) = P(-cold) * P(sneeze|-cold, allergic) * P(allergic|cat) * P(scratches|cat) * P(cat) = 0.95 * 0.7 * 0.75 * 0.5 * 0.02 = **0.005**

c) P(-cold, sneeze, allergic, scratches, cat) / sum_cat (P(-cold, sneeze, allergic, scratches, cat)) = P(-cold) · P(sneeze | -cold) · P(allergic | -cold) · P(scratches | sneeze, cat) · P(cat) / (P(-cold, sneeze, allergic, scratches, cat=T)+P(-cold, sneeze, allergic, scratches, cat=F)) = 0.005 / (0.95 * 0.7 * 0.75 * 0.5 * 0.02 + 0.95 * 0.7 * 0.75 * 0.5 * 0.98) = **0.020**

d) P(cat|scratches) = (P(scratches|cat)*P(cat))/P(scratches) = **0.5*0.2/P(scratches)**

e) **P(scratches) = P(cat)*P(scratches|cat) + P(-cat)*P(scratches|-cat)**

3. PDDL

a) Init(At(Person, Car) ∧ HasKey(Person, Car)
∧ ChargedBattery(Car) ∧ GasInTank(Car))
Goal(Running(Car) ∧ At(Person, Car) ∧ HasKey(Person, Car))
Action(StartCar,
PRECOND: At(Person, Car) ∧ HasKey(Person, Car) ∧ ChargedBattery(Car) ∧ GasInTank(Car)
EFFECT: Running(Car) ∧ At(Person, Car) ∧ HasKey(Person, Car))

b) Fluents:
At(x, l, s): Object x is at location l in situation s.
HasKey(x, y, s): Person x has the key for car y in situation s.
ChargedBattery(x, s): The battery of car x is charged in situation s.
GasInTank(x, s): There is gas in the tank of car x in situation s.
Running(x, s): Car x is running in situation s.

Possibility Axiom:
StartCar(x, s): Try to start car x in situation s.

Preconditions:
At(Person,Car,s)∧HasKey(Person,Car,s)∧ChargedBattery(Car,s)∧GasInTank(Car,s) ⇒Poss(StartCar(Car,s))

Successor State Axiom:

Poss(StartCar(Car,s))⇒(Running(Car,Result(StartCar(Car,s),s))⇔Running(Car,s)
)

Unique Action Axiom

StartCar(…)=/=AnyOtherAction(…):
StartCar(x…x_n)=StartCar(y…y_n)⇔x=y,...,x_n=y_n

Solution:

Running(Car,s)∧At(Person,Car,s)∧HasKey(Person,Car,s)

c) Frame Axiom:

Poss(StartCar(Car1,s)) ⇒ ∀Car2 =
Car1(GasInTank(Car2,Result(StartCar(Car1,s),s) ⇔ GasInTank(Car2,s))