1. Bayesian Inference

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

a) $P(Smart, Study, Pass) = P(Smart) \times P(Study) \times P(Pass|Smart, Study)$

b) JPT Table for Smart, Study, Pass:

	¬smart		smart	
	¬study	study	¬study	study
¬pass	0.336	0.112	0.054	0.006
pass	0.084	0.168	0.126	0.114

c) $P(\neg study, pass) = P(smart, \neg study, pass) + P(\neg smart, \neg study, pass)$
$= 0.126 + 0.084 = 0.21$
$P(smart|\neg study, pass) = P(smart, \neg study, pass) \,/\, P(\neg study, pass)$
$= 0.126 \,/\, 0.21 = 0.6$
$P(smart|\neg study, pass) = 0.6$

d) $P(smart, \neg pass) = P(smart, \neg study, \neg pass) + P(smart, study, \neg pass)$
$= 0.054 + 0.006 = 0.06$
$P(\neg study|smart, \neg pass) = P(smart, \neg study, \neg pass) \,/\, P(smart, \neg pass)$
$= 0.054 \,/\, 0.06 = 0.9$
$P(\neg study|smart, \neg pass) = 0.9$

e) $P(pass|smart) = P(smart, \neg study, pass) + P(smart, study, pass)$
$= 0.126 + 0.114 = 0.24$
$P(pass|smart) = 0.24$

f) $P(pass|study) = P(\neg smart, study, pass) + P(smart, study, pass)$
$= 0.168 + 0.114 = 0.282$
$P(pass|study) = 0.282$

2. Bayesian Networks

a) $P(Cold, Sneeze, Allergic, Scratches, Cat) = P(\phi) = \sum_{\omega \in \phi} P(\omega)$
$P(Cold, Sneeze, Allergic, Scratches, Cat) = P(Cold) \times P(Cat) \times P(Allergic|Cat)$
$\times P(Sneeze|Cold, Allergic) \times P(Scratches|Cat)$

b) $P(\neg cold, sneeze, allergic, scratches, cat) = 0.95 * 0.02 * 0.75 * 0.7 * 0.5$
$= 0.005$

c) $P(cat|\neg cold, sneeze, allergic, scratches) = 0.995$

d) $P(cat|scratches) = 0.1695$

e) $P(scratches) = \sum(P(scratches|cat) \times P(cat)) + \sum(P(scratches|\neg cat) \times P(\neg cat))$

3. PDDL and Situation Calculus

a) PDDL operator:

```
(:action start-car
   :parameters (?car - car ?key - key ?loc - location)
   :precondition (and
                   (at ?car ?loc)
                   (has-key ?key)
                   (battery-charged ?car)
                   (has-gas ?car))
   :effect (and
              (car-running ?car)
              (not (battery-charged ?car))
              (decrease (gas-level ?car) 1))
)
```

b) Situation Calculus:

```
Action: StartCar(c, s)
Preconditions:
   At(c, loc, s) ∧ HasKey(key, s) ∧ BatteryCharged(c, s) ∧ HasGas(c,
s)
Effects:
   CarRunning(c, do(StartCar(c, s))) ∧
   ¬BatteryCharged(c, do(StartCar(c, s))) ∧
   ¬HasGas(c, do(StartCar(c, s)))
```

c) Frame Axiom:

$\forall c,\ c_2,\ s\ (c \neq c_2) \rightarrow (HasGas(c_2,\ s) \leftrightarrow HasGas(c_2,\ do(StartCar(c,\ s))))$