1 Forward chaining

In this section, we will be proving a statement using forward chaining.

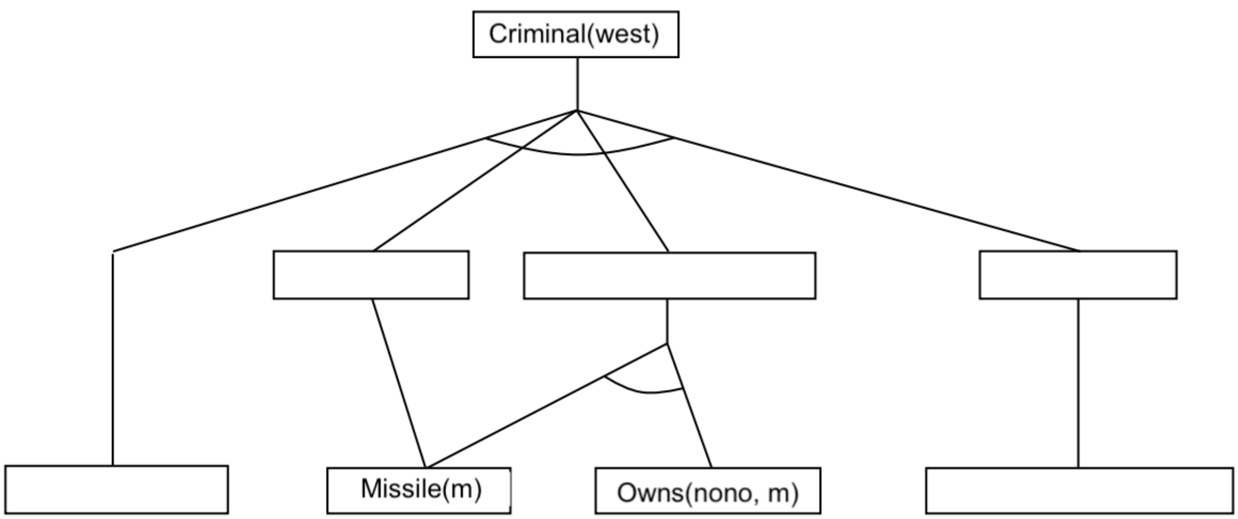
There is currently a war going on and the United States is desperate to round up all the criminals. We want to determine whether Colonel West is a criminal. Let's start with what we know.

We know that it is a crime for an American to sell weapons to hostile nations. The country Nono is an enemy of America. Furthermore, we know that Nono has some missiles, all of which were sold to it by Colonel West, who is American.

(a) Represent your knowledge base using first order logic. You can use the following function predicates: American(x), Criminal(x), Hostile(x), Missile(x), Weapon(x), Enemy(x,y), Owns(x,y), Sells(x,y,z).

1. ______________ ∧ ______________ ∧ ______________ ∧ ______________ ⇒ Criminal(x)
2. Missile(x) ⇒ ______________
3. Missile(m)
4. Owns(nono, m)
5. Missile(x) ∧ ______________ ⇒ Sells(west, x, nono)
6. Enemy(x, america) ⇒ ______________
7. ______________
8. ______________

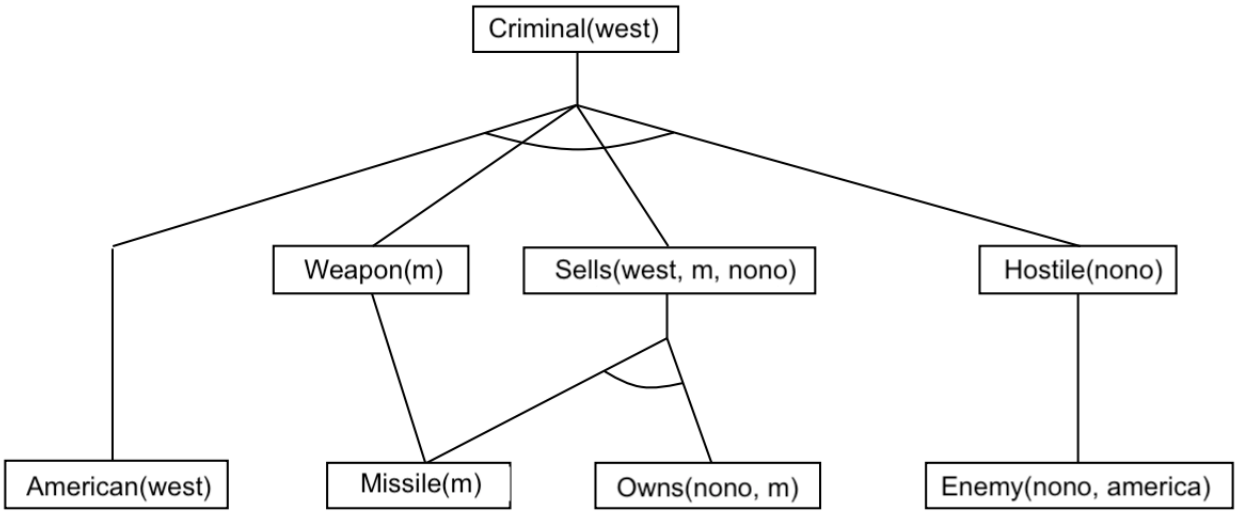
(b) Fill in the blanks below using your knowledge base to prove that Colonel West is a criminal.



(a) Represent your knowledge base using first order logic.

1. American(x) ∧ Weapon(y) ∧ Sells(x,y,z) ∧ Hostile(z) ⇒ Criminal(x)
2. Missile(x) ⇒ Weapon(x)
3. Missile(m)
4. Owns(nono, m)
5. Owns(nono, x) ∧ Missile(x) ⇒ Sells(west, x, nono)
6. Enemy(x, America) ⇒ Hostile(x)
7. American(west)
8. Enemy(nono, america)

(b) Fill in the blanks below using your knowledge base to prove that Colonel West is a criminal.


Criminal(west)
Weapon(m)
Sells(west, m, nono)
Hostile(nono)
American(west)
Missile(m)
Owns(nono, m)
Enemy(nono, america)

2 First-Order Logic

(a) Write in first-order logic the assertion that every key and at least one of every pair of socks will eventually be lost forever, using only the following vocabulary:

- $Key(x), x$ is a key
- $Sock(x), x$ is a sock
- $Pair(x, y), x$ and y are a pair
- Now is the current time
- $Before(t_1, t_2)$ represents that time t_1 comes before t_2
- $Lost(x, t)$ represents that object x is lost at time t

$\forall k\ Key(k) \Rightarrow [\exists t_0\ Before(Now, t_0) \wedge \forall t\ Before(t_0, t) \Rightarrow Lost(k, t)]$
$\forall s_1, s_2\ Sock(s_1) \wedge Sock(s_2) \wedge Pair(s_1, s_2) \Rightarrow$
$[\exists t_1\ Before(Now, t_1) \wedge \forall t\ Before(t_1, t) \Rightarrow Lost(s_1, t)] \vee$
$[\exists t_2\ Before(Now, t_2) \wedge \forall t\ Before(t_2, t) \Rightarrow Lost(s_2, t)]$

Notice that the disjunction allows for both socks in the pair to be lost, as the English sentence implies.

(b) Write out the vocabulary you would use to represent the following sentence in first-order logic.

- Everyone who takes 15-281 loves Pacman.

  Student(x), x is a student
  In281(x), x is in 15-281
  Loves(x, y), x loves y
  Pacman(x), x is a Pacman

3 Cargo Plane: Linear Programming Formulation

A cargo plane has three compartments for storing cargo: front, centre and rear. These compartments have the following limits on both weight and space:

Compartment	Weight capacity (tonnes)	Space capacity (cubic metres)
Front	10	6800
Centre	16	8700
Rear	8	5300

The following four cargoes are available for shipment on the next flight:

Cargo	Weight (tonnes)	Volume (cubic metres/tonne)	Profit ($/tonne)
C1	18	480	310
C2	15	650	380
C3	23	580	350
C4	12	390	285

Any proportion of these cargoes can be accepted. The objective is to determine how much of each cargo C1, C2, C3 and C4 should be accepted and how to distribute each among the compartments so that the total profit for the flight is maximised. **Formulate** the above problem as a linear program (what is the objective and the constraints?). Think about the assumptions you are making when formulating this problem as a linear program.

Variables:

We need to decide how much of each of the four cargoes to put in each of the three compartments. Hence let $x_{i,j}$ be the number of tonnes of cargo i (i=1,2,3,4 for C1, C2, C3 and C4 respectively) that is put into compartment j (j=1 for Front, j=2 for Centre and j=3 for Rear) where $x_{i,j} \geq 0; i = 1, 2, 3, 4; j = 1, 2, 3$.

(Note here that we are explicitly told we can split the cargoes into any proportions (fractions) that we like.)

Constraints:

1. We cannot pack more of each of the four cargoes than we have available.

$$x_{1,1} + x_{1,2} + x_{1,3} \leq 18$$

$$x_{2,1} + x_{2,2} + x_{2,3} \leq 15$$

$$x_{3,1} + x_{3,2} + x_{3,3} \leq 23$$

$$x_{4,1} + x_{4,2} + x_{4,3} \leq 12$$

2. The weight capacity of each compartment must be respected.

$$x_{1,1} + x_{2,1} + x_{3,1} + x_{4,1} \leq 10$$

$$x_{1,2} + x_{2,2} + x_{3,2} + x_{4,2} \leq 16$$

$$x_{1,3} + x_{2,3} + x_{3,3} + x_{4,3} \leq 8$$

3. The volume (space) capacity of each compartment must be respected.

$$480x_{1,1} + 650x_{2,1} + 580x_{3,1} + 390x_{4,1} \leq 6800$$

$$480x_{1,2} + 650x_{2,2} + 580x_{3,2} + 390x_{4,2} \leq 8700$$

$$480x_{1,3} + 650x_{2,3} + 580x_{3,3} + 390x_{4,3} \leq 5300$$

Objective: The objective is to maximise total profit, i.e.

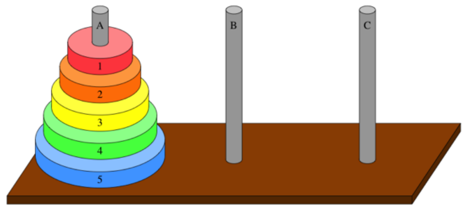
maximise $310(x_{1,1} + x_{1,2} + x_{1,3}) + 380(x_{2,1} + x_{2,2} + x_{2,3}) + 350(x_{3,1} + x_{3,2} + x_{3,3}) + 285(x_{4,1} + x_{4,2} + x_{4,3})$

The basic assumptions are:

1. that each cargo can be split into whatever proportions/fractions we desire
2. that each cargo can be split between two or more compartments if we so desire
3. that the cargo can be packed into each compartment (for example if the cargo was spherical it would not be possible to pack a compartment to volume capacity, some free space is inevitable in sphere packing)
4. all the data/numbers given are accurate

4 Planning Tower of Hanoi

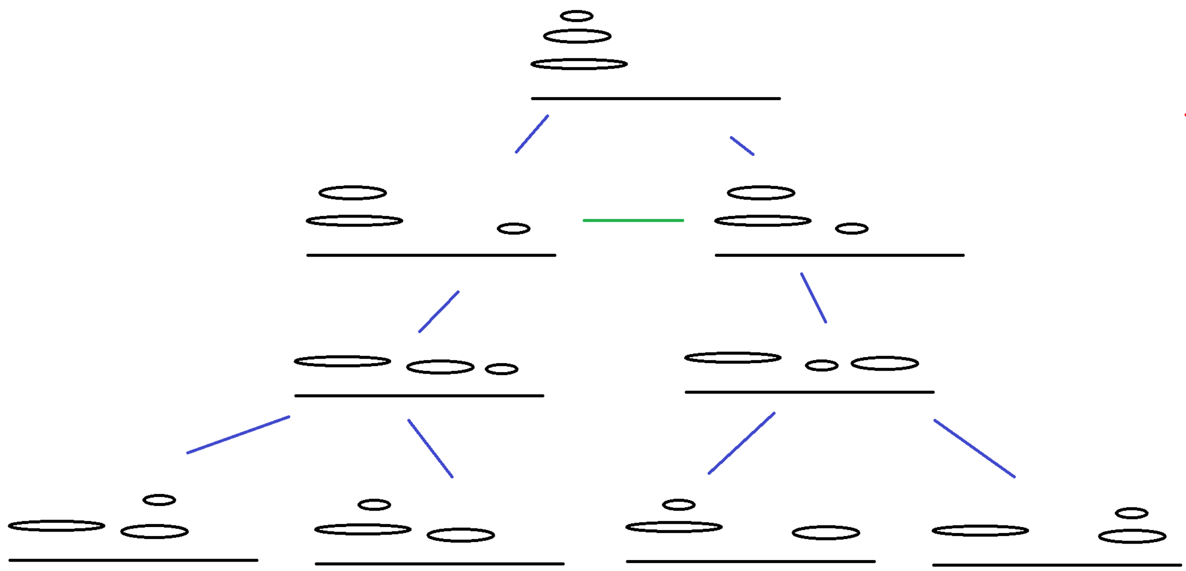
In the Tower of Hanoi problem, you are given n disks, each of a distinct size, and 3 rods, A, B and C. The disks start off stacked on top of each other on rod A, stacked from largest being the lowest to smallest being the highest in a "tower", and the goal is to move that tower to the rod C. You can only move a disk to an empty rod or on top of a larger disks, and disks may only have one other disk on its surface (they must be stacked linearly).



(a) Assume we have 3 disks. Formulate the problem as a graph-planning problem, specifying instances, operators, and start/goal states.

See provided sample code

(b) Draw the planning graph for the first 3 moves. You may use pictures instead of propositions.



(c) Generalize the problem formulation for n disks.

See provided sample code