1.

∀x (bowlingBall(x) → sportingEquipment(x))

∀h ∀f (horse(h) ∧ frog(f) → fasterThan(h,f))

∀h (domesticatedHorse(h) → ∃o owner(o, h))

∀h ∀r (horse(h) ∧ rider(r, h) → ∃o (owner(o, h) ∧ ¬(r = o)))

∀x (finger(x) → ∃y (hand(y) ∧ digitOf(x, y) ∧ ¬thumb(x)))

∀t (isoscelesTriangle(t) → ∃e1 ∃e2 ∃e3 (edge(e1, t) ∧ edge(e2, t) ∧ edge(e3, t) ∧ sameLength(e1, e2) ∧ ¬sameLength(e1, e3)))

2.

* ¬ person(*x*)∨¬ petOf(*x*,*c*)∨ doglover(*x*)
* ¬ person(*x*)∨¬ petOf(*x*,*f*(*x*))∨ doglover(*x*)
* ¬ person(*x*)∨ dog(*c*)∨ doglover(*x*)

3.

1. owes(owner(X),citibank,cost(X)) and owes(owner(ferrari),Z,cost(Y))  
 Unifier: {X/ferrari, Z/citibank, Y/X}  
 Unified: owes(owner(ferrari),citibank,cost(ferrari))

2. gives(bill, jerry, book21) and gives(X,brother(X),Z)  
 Not unifiable because 'bill' cannot be unified with 'X' and 'jerry' cannot be unified with 'brother(X)'

3. opened(X,result(open(X),s0)) and opened(toolbox,Z)  
 Not unifiable because 'result(open(X),s0)' cannot be unified with any single variable 'Z'.

4.

First-Order Logic Translation

1. Pompeian(Marcus).
2. ∀x (Pompeian(x) → Roman(x)).
3. Ruler(Caesar).
4. ∀x (Roman(x) → (LoyalTo(x, Caesar) ⊕ Hate(x, Caesar))).
5. ∀x ∃y (LoyalTo(x, y)).
6. ∀x ∀y ((Assassinate(x, y) ∧ Ruler(y)) → ¬LoyalTo(x, y)).
7. Assassinate(Marcus, Caesar).

Natural Deduction Proof

One can only attempt to assassinate a ruler they are not loyal to. Thus, by the exclusive or condition for Romans, Marcus must hate Caesar.

1. Marcus is pompeian (1)
2. Marcus is roman (2)
3. Caesar is a ruler (3)
4. Marcus is either loyal to or hate caesar (2) (5)
5. If you try to assasinate a ruler you are not loyal to them (7)
6. Marcus tried to assasinate caesar (7)
7. Marcus is not loyal to caesar (4)

Conversion to CNF

1. Pompeian(Marcus).  
 2. ∀x (¬Pompeian(x) ∨ Roman(x)).  
 3. Ruler(Caesar).  
 4. ∀x (¬Roman(x) ∨ LoyalTo(x, Caesar) ∨ Hate(x, Caesar)) and ∀x (¬Roman(x) ∨ ¬(LoyalTo(x, Caesar) ∧ Hate(x, Caesar))).  
 5. ∀x LoyalTo(x, f(x)) (where f(x) is a Skolem function).  
 6. ∀x ∀y (¬Assassinate(x, y) ∨ ¬Ruler(y) ∨ ¬LoyalTo(x, y)).  
 7. Assassinate(Marcus, Caesar).

Resolution Refutation Proof

1. ¬Hate(Marcus,Caesar)
2. ¬Roman(*Marcus*)∨LoyalTo(*Marcus*,*Caesar*)
3. Roman(Marcus) (Given)
4. LoyalTo(*Marcus*,*Caesar*)
5. ∀x ∀y (¬Assassinate(Marcus, Caesar) ∨ ¬Ruler(Caesar))
6. ∀x ∀y (¬Assassinate(Marcus, Caesar)) (because caesar is a ruler)
7. Empty clause because Assassinate(Marcus, Caesar). Invalidates ¬Hate(Marcus,Caesar) meaning Hate(Marcus,Caesar)

5.

**Map-coloring**

Every state has exactly one color:

∀s ∃!c color(s, c)

Adjacent states must have different colors:

∀s1, s2, c (adjacent(s1, s2) ∧ color(s1, c) → ¬color(s2, c))

**Sammy's Sport Shop**

If a box is observed with tennis balls of a certain color, then it contains only tennis balls of that color:

∀x, q (obs(x, q) → cont(x, q))

If a box is labeled with a certain color, then it contains only tennis balls of that color:

∀x, q (label(x, q) → cont(x, q))

**Wumpus World**

Defining adjacent rooms:

adjacent(x, y, p, q) as an existing predicate.

Rules for perceptions (stench, breezy, safe):

∀x, y (stench(x, y) → ∃p, q (adjacent(x, y, p, q) ∧ wumpus(p, q)))

∀x, y (breezy(x, y) → ∃p, q (adjacent(x, y, p, q) ∧ pit(p, q)))

∀x, y (safe(x, y) → ¬wumpus(x, y) ∧ ¬pit(x, y))

**4-Queens**

No two queens can be in the same row or column:

∀r, c, r', c' ((queen(r, c) ∧ queen(r', c')) → (r ≠ r' ∧ c ≠ c'))

No two queens can be on the same diagonal:

∀r, c, r', c' ((queen(r, c) ∧ queen(r', c') ∧ abs(r - r') = abs(c - c')) → (r = r' ∧ c = c'))