**Problem Formulation**

1. State
   1. Informally: The items in the fridge
   2. Formally (mathematically)
      1. S = {(x0, x1, …, xm - 1) | x0, x1, …, xm - 1 <= y, x0 + x1 + … + xm – 1 <= z}
      2. Where m = the number of different items, y = maximum allowed number of each item and z = the capacity of the fridge
2. Action
   1. Informally: Purchase items
   2. Formally (mathematically)
      1. A = {(x0, x1, …, xm - 1) | x0, x1, …, xm - 1 <= y, x0 + x1 + … + xm – 1 <= n}
      2. Where m = the number of different items, y = maximum allowed number of each item and n = the maximum number of items we can order
3. Transition
   1. Informally: At the beginning of each week, given the current inventory of our fridge, we purchase food and at the end of the week, we are left with the number of items not consumed in the fridge.
   2. Formally: (mathematically)
      1. T(s, a, s’) = P(St + 1 = s’ | St = s, At = a) where s, s’ are elements of S and a is an element of A
4. Reward
   1. Informally: Given the current state, the reward will be the penalty in the current week multiplied by the discount factor, which is raised to the power of the number representing the current week
   2. Formally: (mathematically)
      1. R(s, a) = xh \* PenaltyAtWeek(h) where x = discount factor and h is the number representing the current week and 0 <= h <= N where N is the maximum number of weeks we are accounting for.
   3. PenaltyAtWeek(h) is another function which calculates the penalty based on a pre-defined cost and the number of failures for that week.
      1. PenaltyAtWeek(h) = c \* NumOfFailures(h) where c = the pre-defined cost.
   4. NumOfFailures(h) is the number of failures for week-h. The number of failures is calculated based on the state s’ that is a result of the action a taken in state s and the consumption of the user.

**Conceptual Arguments**

My chin is a triangle. Your argument is invalid

**Description of Method**

Buy your own food

**The Level Our Method Solves and Why**

Our method solves a level that no one can make it to. How can we possibly explain that?