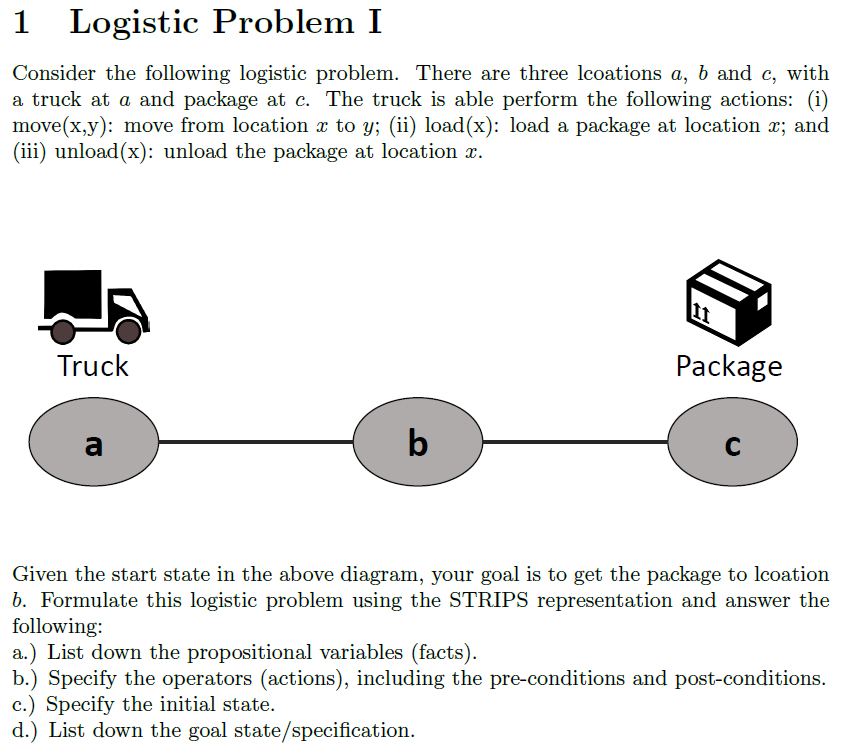
# Logistic Problem 1



## STRIPS Predicate

* truckAt(location)
* at(location, package)
* free(truck)
* carry(package, truck)
* canMove(locX, locY)

## STRIPS Operators

* move(locX, locY)
  + pre: truckAt(locX), canMove(locX, locY)
  + add: truckAt(locY)
  + del: truckAt(locX)
* load(locX)
  + pre: at(locX, package), truckAt(locX), free(truck)
  + add: carry(package, truck)
  + del: free(truck), at(locX, package)
* unload(locY)
  + pre: truckAt(locY), carry(package, truck)
  + add: at(locY, package), free(truck)
  + del: carry(package, truck)

## STRIPS Initial State

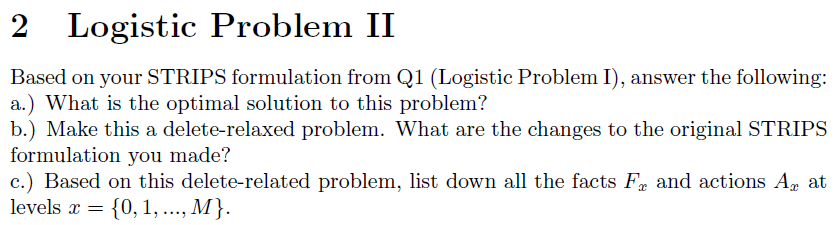
{ truckAt(a), free(truck), at(c, package),

canMove(a, b), canMove(b, a), canMove(b, c), canMove(c, b) }

## STRIPS Goal State

{ at(b, package) }

# Logistic Problem 2



## Optimal Solution

[ move(a, b), move(b, c), load(c), move(c, b), unload(b) ]

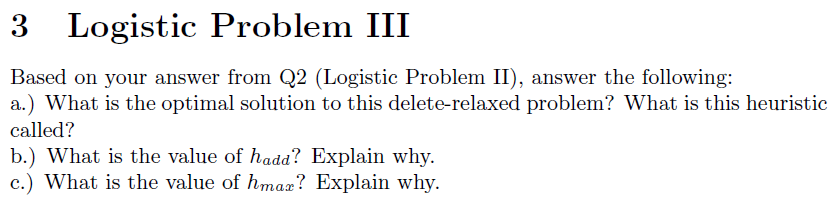
## Delete Relaxed Problem

* move(locX, locY)
  + pre: truckAt(locX), canMove(locX, locY)
  + add: truckAt(locY)
  + ~~del: truckAt(locX)~~
* load(locX)
  + pre: at(locX, package), truckAt(locX), free(truck)
  + add: carry(package, truck)
  + ~~del: free(truck), at(locX, package)~~
* unload(locY)
  + pre: truckAt(locY), carry(package, truck)
  + add: at(locY, package), free(truck)
  + ~~del: carry(package, truck)~~

## Delete Relaxed F & A

* F0 = truckAt(a), free(truck), at(c, package), canMove(a, b), canMove(b, a), canMove(b, c), canMove(c, b)
* A0 = move(a, b)
* F1 = truckAt(a), truckAt(b), free(truck), at(c, package), canMove(a, b), canMove(b, a), canMove(b, c), canMove(c, b)
* A1 = move(b, c)
* F2 = truckAt(a), truckAt(b), truckAt(c), free(truck), at(c, package), canMove(a, b), canMove(b, a), canMove(b, c), canMove(c, b)
* A2 = load(c)
* F3 = truckAt(a), truckAt(b), truckAt(c), carry(package, truck), free(truck), at(c, package), canMove(a, b), canMove(b, a), canMove(b, c), canMove(c, b)
* A3 = unload(b)
* F4 = truckAt(a), truckAt(b), truckAt(c), carry(package, truck), free(truck), at(c, package), at(b, package), canMove(a, b), canMove(b, a), canMove(b, c), canMove(c, b) #GOAL

# Logistic Problem 3



## Delete Relaxed Optimal Solution

[ move(a, b), move(b, c), load(c), unload(b) ], h+ heuristic

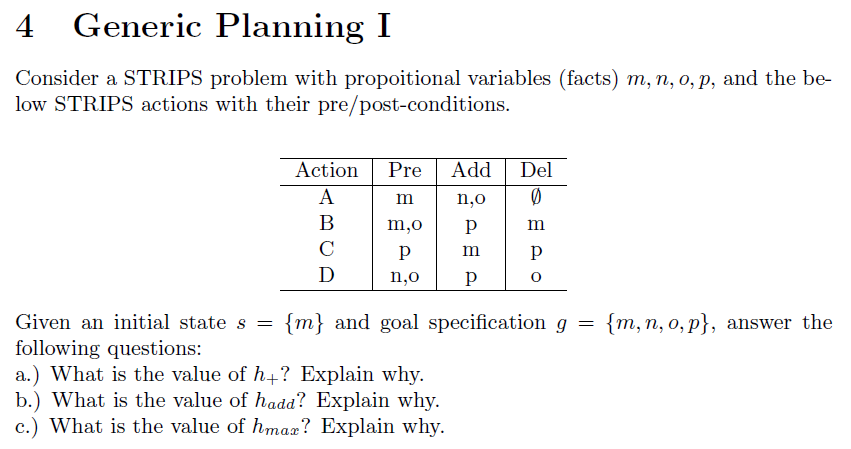
## Value of h-add

h-add = 4 (only 1 goal fact, which is at(b, package) which happens at F4)

## Value of h-max

h-max = 4 (only 1 goal fact, which is at(b, package) which happens at F4)

# Generic Planning 1



* F0 = m
* A0 = A
* F1 = m, n, o
* A1 = B, D
* F2 = m, n, o, p

## Value of h+

h+ = 2 (min number of required to achieve all, which happens at F2)

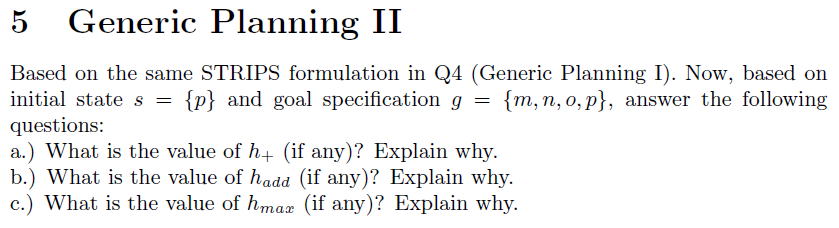
## Value of h-add

h-add = 4 (add cost of m, n, o, p = 0 + 1 + 1 + 2 = 4)

## Value of h-max

h-max = 2 (max cost of m, n, o, p which is p at F2)

# Generic Planning 2



* F0 = p
* A0 = C
* F1 = p, m
* A1 = A
* F2 = m, n, o, p

## Value of h+

h+ = 2 (min number of required to achieve all, which happens at F2)

## Value of h-add

h-add = 5 (add cost of m, n, o, p = 1 + 2 + 2 + 0 = 5)

## Value of h-max

h-max = 2 (max cost of m, n, o, p, which is n and o at F2)