Logan Esch
Artificial Intelligence
Homework #4

1a.

Because the preconditions for an action state the conditions that must be true for that action to be executed, ignoring them will have no effect. This is because the actions will only affect the parts of the state that were covered by the preconditions. For example, given the action Fly(P1, JFK, SFO) with the precondition At(P1, JFK) the action would still result in the plane flying to SFO even if it wasn't at JFK.

1b. Yes. Because the illegal actions will have no effect on the plan, the goal will still be reached.

1c. Yes. Because the illegal actions will have no effect on the plan, the goal will still be reached.

2. I am assuming the goal is for the monkey to be holding the bananas.

Name: Go(object, origin, destination)
Preconditions: at(object, origin) ∧ isMonkey(object) ∧ height(object, elevation) ∧ isLow(elevation)
Add: at(object, destination)
Delete: at(object, origin)

Name: PushAction(actor, object, origin, destination)
Preconditions: at(actor, origin) ∧ at(object, origin) ∧ isMonkey(actor) ∧ height(object, elevation) ∧ height(actor, elevation) ∧ isLow(elevation)
Add: at(object, destination), at(actor, destination), climbableHight(destination, High)
Delete: at(object, origin), at(actor, origin), climbableHight(origin, Low)

Name: Climb(actor, place)
Preconditions: at(actor, place) ∧ height(actor, elevation) ∧ isMonkey(actor) ∧ isLow(elevation) ∧ climbableHight(place, High)
Add: height(actor, High)
Delete: height(actor, Low)

Name: Grasp(actor, object)
Preconditions: isMonkey(actor) ∧ at(actor, place) ∧ at(object, place), height(actor, elevation) ∧ height(object, elevation)
Add: holding(actor, object)

Initial State:
at(Monkey, A)
at(Bananas, B)
at(Box, C)
height(Monkey, Low)
height(Bananas, High)
height(Box, Low)
climbableHight(A, Low)

climbableHight(B, Low)
climbableHight(C, High)
holding()

Goal Stack:
holding(Bananas)

Pop: holding(Bananas)
goal satisfied: False
Push: holding(Bananas)
Push: Grasp(Monkey, Bananas), Achieve(isMonkey(Monkey)), Achieve(at(Monkey, B), Achieve(at(Bananas, B), Achieve(height(Monkey, High)), Achieve(height(Bananas, High))

Pop: Achieve(height(Bananas, High))
goal satisfied: True

Pop: Achieve(height(Monkey, High))
goal satisfied: False
Push: Achieve(height(Monkey, High))
Push: Climb(Monkey, B), Achieve(at(Monkey, B)), Achieve(height(Monkey, Low)), Achieve(isMonkey(Monkey)), Achieve(isLow(Low)), Achieve(climbableHight(B, High))

Pop: Achieve(climbableHight(B, High))
goal satisfied: False
Push: Achieve(climbableHight(B, High))
Push: PushAction(Monkey, Box, C, B), Achieve(at(Monkey, C)), Achieve(at(Box, C)), Achieve(isMonkey(Monkey)), Achieve(height(Box, Low)), Achieve(height(Monkey, Low)) Achieve(isLow(Low))

//saving space:
Pop: Achieve(isLow(Low)), Achieve(height(Monkey, Low)), Achieve(height(Box, Low)), Achieve(isMonkey(Monkey)), Achieve(at(Box, C))
Goals satisfied: True

Pop: Achieve(At(Monkey, C)
goal satisfied: False
Push: Achieve(At(Monkey, C)
Push: Go(Monkey, A, C), Achieve(at(Monkey, A)), Achieve(isMonkey(Monkey)), Achieve(height(Monkey, Low)), Achieve(isLow(Low))

//Saving space:
Pop: Achieve(isLow(Low)), Achieve(height(Monkey, Low)), Achieve(isMonkey(Monkey)), Achieve(at(Monkey, A))
goals satisfied: True

Pop: Go(Monkey, A, C)
Plan: Go(Monkey, A, C),
State:

at(Monkey, C)
at(Bananas, B)
at(Box, C)
height(Monkey, Low)
height(Bananas, High)
height(Box, Low)
climbableHight(A, Low)
climbableHight(B, Low)
climbableHight(C, High)
holding()

Goal Stack:

Achieve(At(Monkey, C)
PushAction(Monkey, Box, C, B)
Achieve(climbableHight(B, High))
Achieve(isLow(Low))
Achieve(isMonkey(Monkey))
Achieve(height(Monkey, Low))
Achieve(at(Monkey, B))
Climb(Monkey, B)
Achieve(height(Monkey, High))
Achieve(at(Bananas, B)
Achieve(at(Monkey, B)
Achieve(isMonkey(Monkey))
Grasp(Monkey, Bananas)

Pop: Achieve(At(Monkey, C)
goal satisfied: True

Pop: PushAction(Monkey, Box, C, B)
Plan: Go(Monkey, A, C), PushAction(Monkey, Box, C, B)
State:

at(Monkey, B)
at(Bananas, B)
at(Box, B)
height(Monkey, Low)
height(Bananas, High)
height(Box, Low)
climbableHight(A, Low)
climbableHight(B, High)

climbableHight(C, Low)
holding()

Goal Stack:

Achieve(climbableHight(B, High))
Achieve(isLow(Low))
Achieve(isMonkey(Monkey))
Achieve(height(Monkey, Low))
Achieve(at(Monkey, B))
Climb(Monkey, B)
Achieve(height(Monkey, High))
Achieve(at(Bananas, B)
Achieve(at(Monkey, B)
Achieve(isMonkey(Monkey))
Grasp(Monkey, Bananas)

//condensed:
Pop: Achieve(climbableHight(B, High)), Achieve(isLow(Low)), Achieve(isMonkey(Monkey)), Achieve(height(Monkey, Low)), Achieve(at(Monkey, B))
goals satisfied: True

Pop: Climb(Monkey, B)
Plan: Go(Monkey, A, C), PushAction(Monkey, Box, C, B), Climb(Monkey, B)

State:

at(Monkey, B)
at(Bananas, B)
at(Box, B)
height(Monkey, High)
height(Bananas, High)
height(Box, Low)
climbableHight(A, Low)
climbableHight(B, High)
climbableHight(C, Low)
holding()

Goal Stack:

Achieve(height(Monkey, High))
Achieve(at(Bananas, B)
Achieve(at(Monkey, B)
Achieve(isMonkey(Monkey))
Grasp(Monkey, Bananas)

//condensed:
Pop: Achieve(height(Monkey, High)), Achieve(at(Bananas, B), Achieve(at(Monkey, B), Achieve(isMonkey(Monkey)), Grasp(Monkey, Bananas)
goals satisfied: True

Pop: Grasp(Monkey, Bananas)
Plan: Go(Monkey, A, C), PushAction(Monkey, Box, C, B), Climb(Monkey, B), Grasp(Monkey, Bananas)
State:

at(Monkey, B)
at(Bananas, B)
at(Box, B)
height(Monkey, High)
height(Bananas, High)
height(Box, Low)
climbableHight(A, Low)
climbableHight(B, High)
climbableHight(C, Low)
holding(Monkey, Bananas)

Goal stack:

//empty, return pan.

Plan: Go(Monkey, A, C), PushAction(Monkey, Box, C, B), Climb(Monkey, B), Grasp(Monkey, Bananas)

3a.

Initial State: Atrobot(B), At(P1, B), At(P2, B), At(P3, B), Carrying(), Fueled(False), Cango(B, S), CanGo(S, M)
Goal: Atrobot(M)

3b.

Name: Goto(x, y)
Preconditions: AtRobot(x) ∧ Fueled(True) ∧ CanGo(x, y)
Add: AtRobot(y)
Delete: AtRobot(x), Fueled(False)

Name: Pickup(u, x)
Preconditions: AtRobot(x) ∧ At(u, x) ∧ Carrying()
Add: Carrying(u)
Delete: At(u, x), Carrying()

Name: Putdown(u, x)
Preconditions: AtRobot(x) ∧ Carrying(u)
Add: At(u, x), Carrying()
Delete: Carrying(u)

Name(Refuel, u, x)
Preconditions: AtRobot(x) ∧ Fueled(False) ∧ At(u, x)
Add: Fueled(True)
Delete: Fueled(False), At(u,x)

3c.

Atrobot(B), At(P1, B), At(P2, B), At(P3, B), Carrying(), Fueled(False), Cango(B, S), CanGo(S, M)
Refuel(P1, B)
Atrobot(B), At(P2, B), At(P3, B), Carrying(), Fueled(True), Cango(B, S), CanGo(S, M)
Pickup(P2, B)
Atrobot(B), At(P3, B), Carrying(P2), Fueled(True), Cango(B, S), CanGo(S, M)
Goto(S)
Atrobot(S), At(P3, B), Carrying(P2), Fueled(False), Cango(B, S), CanGo(S, M)
Putdown(P2)
Atrobot(S), At(S, P2), At(P3, B), Carrying(), Fueled(False), Cango(B, S), CanGo(S, M)
Refuel(P2, S)
Atrobot(S), At(P3, B), Carrying(), Fueled(True), Cango(B, S), CanGo(S, M)
GoTo(M)
Atrobot(M), At(P3, B), Carrying(), Fueled(False), Cango(B, S), CanGo(S, M)

4.

S0	A0	S1	A1	S2	A2	S3	A3	S4
On(Sock, Left), False	Puton(Sock, Left)	On(Sock, Left), True	Puton(Sock, Right)	On(Sock, Left), True On(Sock, Right), True	Puton(Shoe, Left)	On(Sock, Left), True On(Sock, Right), True	Puton(Shoe, Right)	
On(Sock, Right), False	Puton(Sock, Right)	On(Sock, Right), True	Puton(Sock, Left)					

On(Shoe, Left), False	Puton(Shoe, Left)	On(Shoe, Left), True	Puton(Shoe, Left)					
On(Shoe, Right), False	Puton(Shoe, Right)	On(Shoe, Right), True,	Puton(Shoe, Right)					

5a. The only threat arcs that need to be added to this partial order plan are between Optimize and Debug, as optimizing software could introduce bugs. Shipping software would not affect any of the other conditions.

5b. Optimize has the precondition Have Program that is not supported by a causal link.
Design Packaging is not in the plan and also has the precondition of Have Program that is not supported by a causal link.

5c.