FIT5047-Assignment-1

Name: Huan Yang

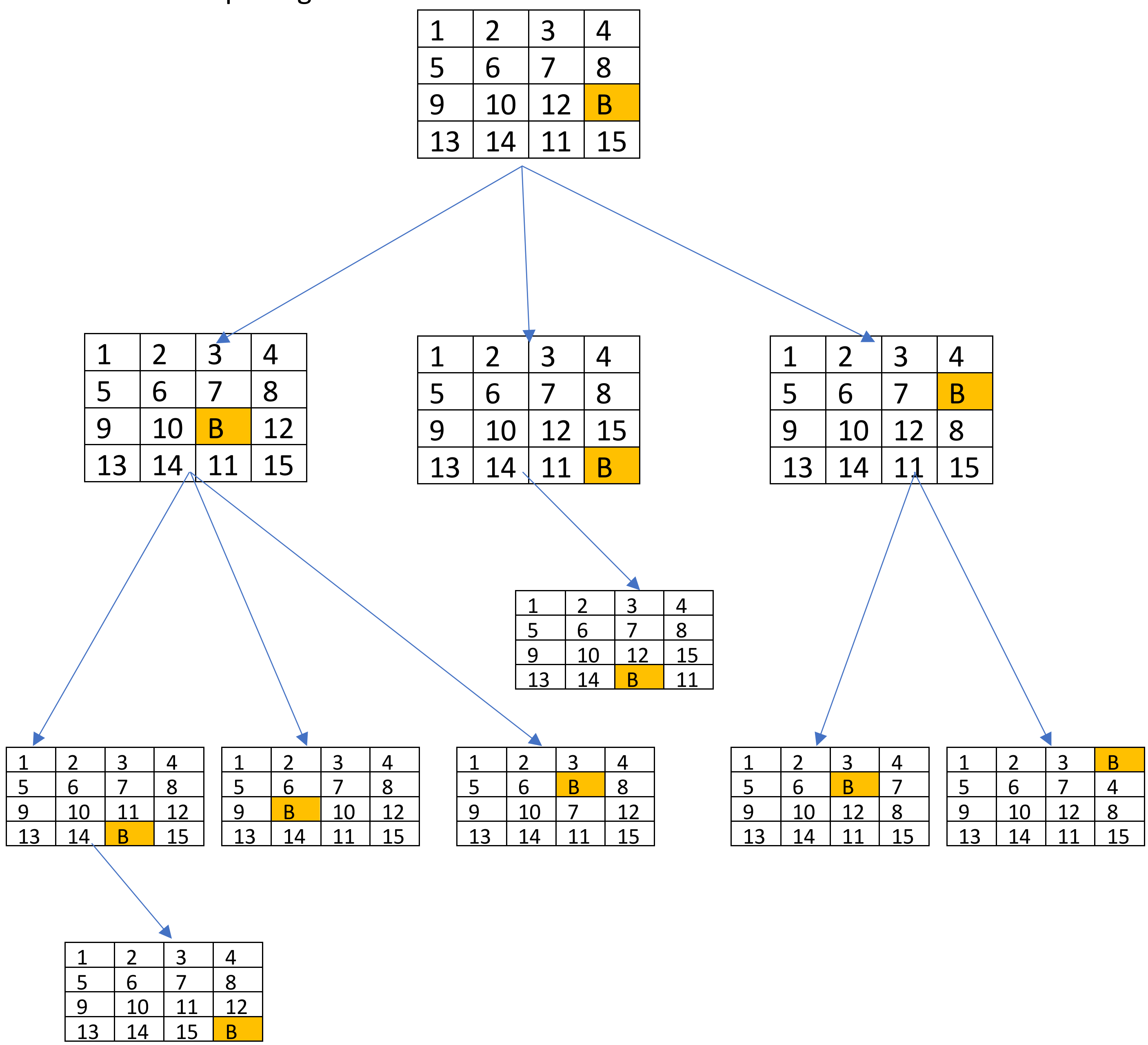
Student Number: 27620042

Question 1

1a

h1 = 3 (How many puzzles are not in the initial state)
h2 = 3 (How many steps do we need to take at least)
h3 = 9 (How many steps do we need to take at most)

1b Iterative Deepening



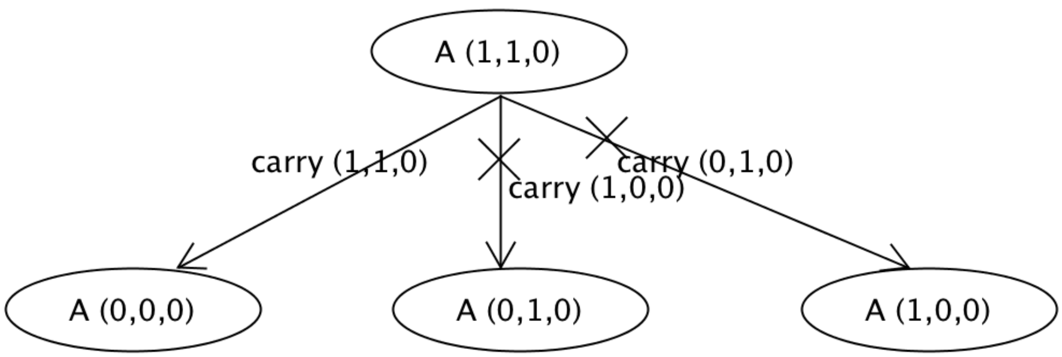
We do not consider the steps which going back. Because the cycle is meaningless.

1c Please see the code

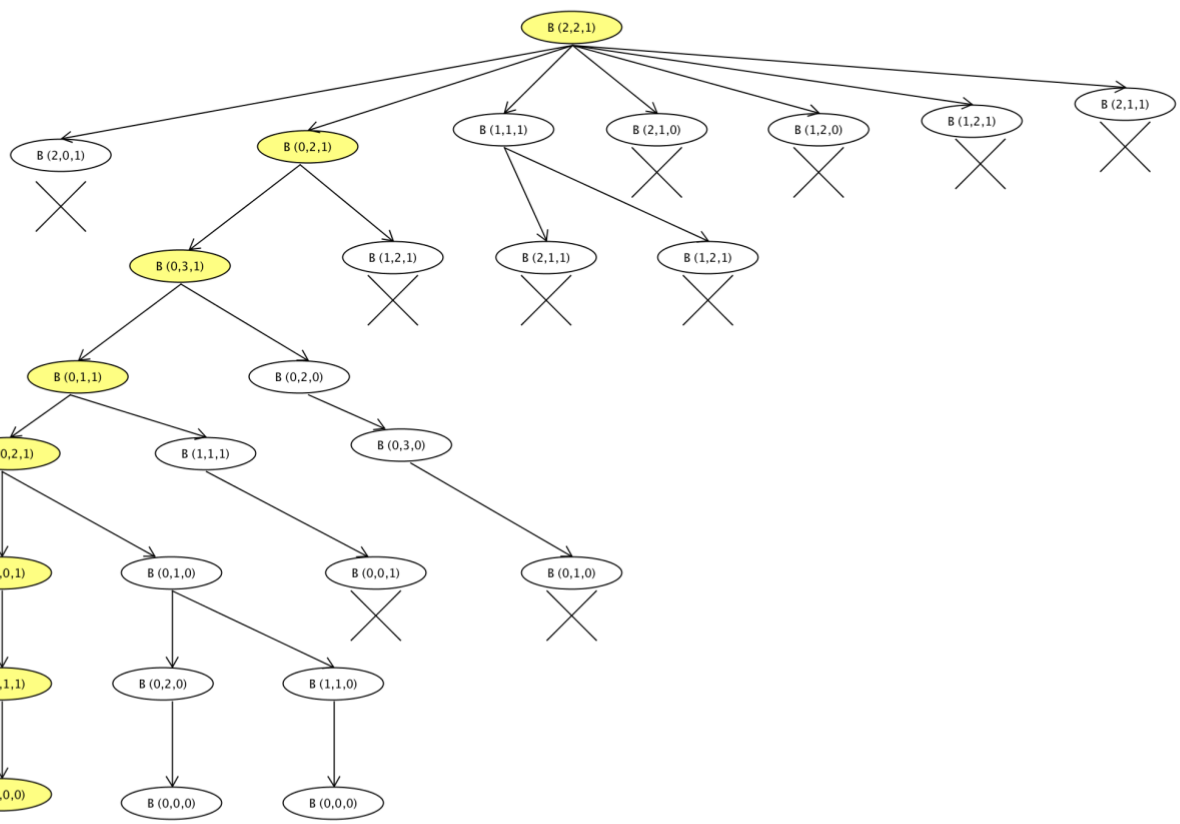
Question 2

The most efficient delivery option is carrying 2 passengers to the next point and return 1. Thus, I will ignore the situations which are bringing 2 passengers to the next point and return 2, and bringing 1 passengers to the next point and return 1 or 2.

A ——>B or C
Origin State: (1,1,0) = (Mice, Cats, 15-puzzle)
Goal State: (0,0,0)
Operators: Carry (1,1,0) Carry (1,0,0) Carry (0,1,0)
The most efficient step = 1



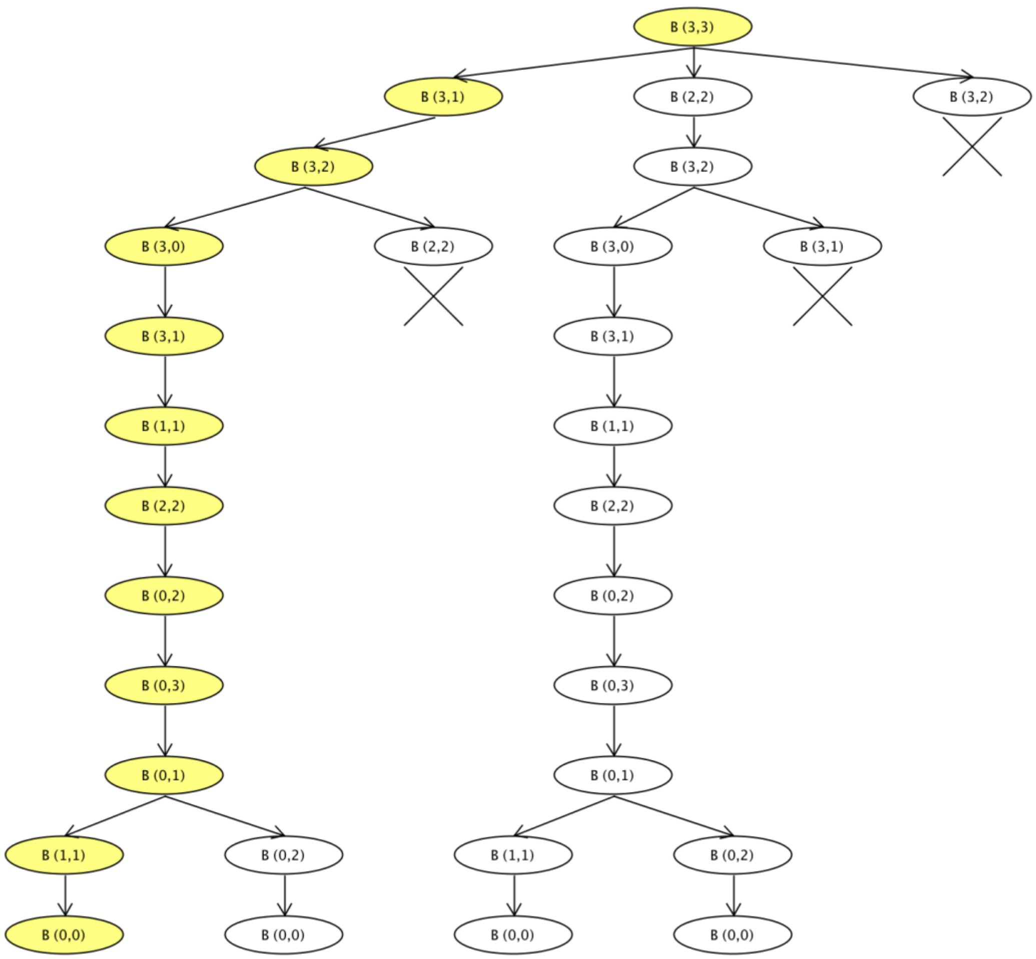
B ——>C v1.0
Origin State: (2,2,1) = (Mice, Cats, 15-puzzle)
Goal State: (0,0,0)
Operators: Carry (1,1,0) Carry (1,0,1) Carry (0,1,1) Carry (2,0,0) Carry (0,2,0)



Solution:
Carry (2,0,0) -> Goal
Carry (0,1,0) -> Origin
Carry (0,2,0) -> Goal
Carry (0,1,0) -> Origin
Carry (0,2,0) -> Goal
Carry (0,1,0) -> Origin
Carry (0,1,1) -> Goal

The most efficient step = 7

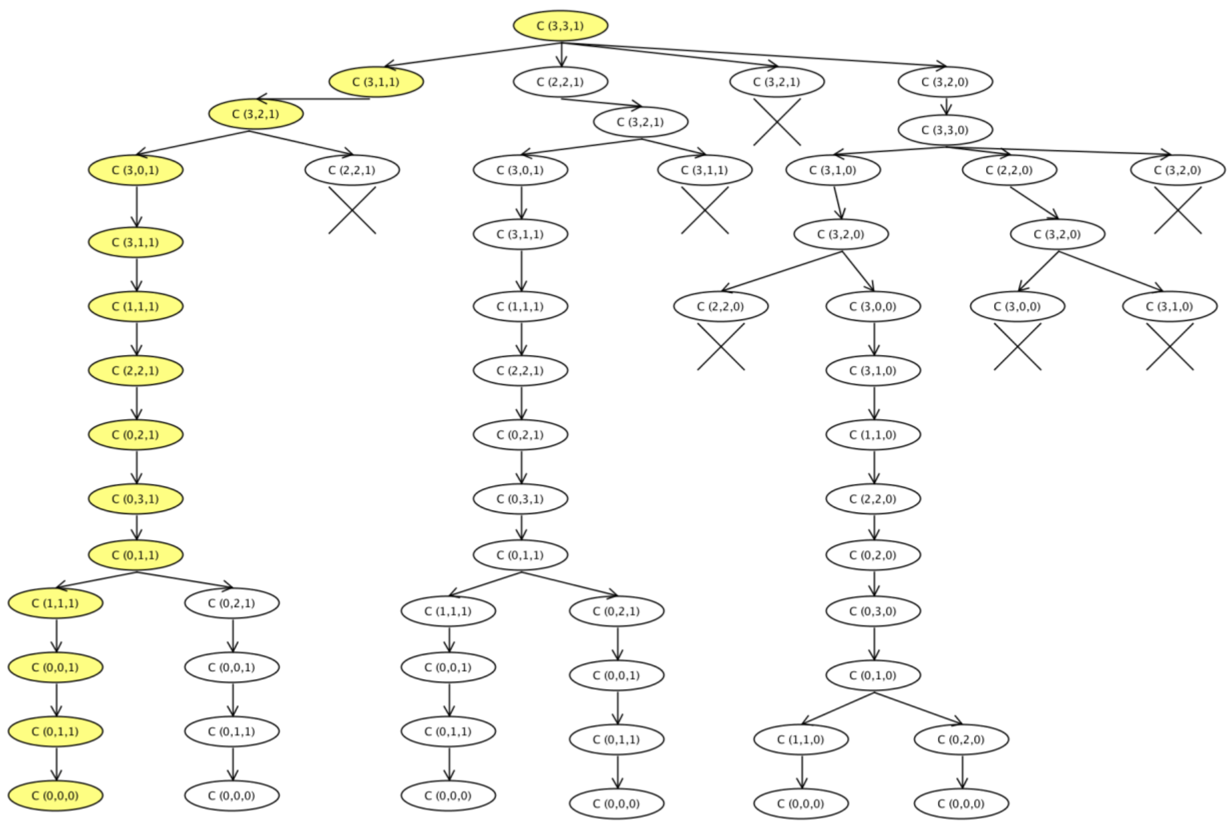
B ——>C v2.0
Origin State: (3,3,0) = (Mice, Cats, 15-puzzle)
Goal State: (0,0,0)
Operators: Carry (1,1,0) Carry (1,0,0) Carry (0,1,0) Carry (2,0,0) Carry (0,2,0)
In this case, I elect instead to have one extra mouse and one extra cat that we have to take with us when we leave B.



Solution:
Carry (0,2) -> Goal
Carry (0,1) -> Origin
Carry (0,2) -> Goal
Carry (0,1) -> Origin
Carry (2,0) -> Goal
Carry (1,1) -> Origin
Carry (2,0) -> Goal
Carry (0,1) -> Origin
Carry (0,2) -> Goal
Carry (1,0) -> Origin
Carry (1,1) -> Goal

The most efficient step = 11
The steps of version 1 of B to C is smaller than version 2. Therefore, I choose version 1

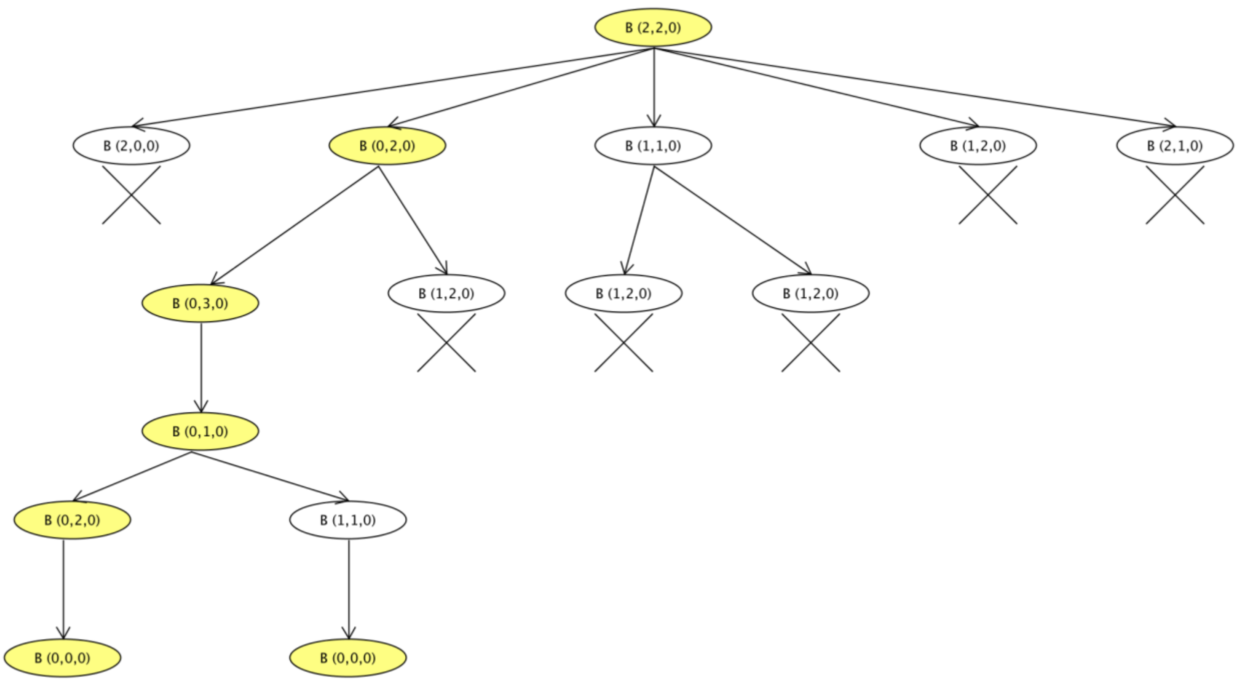
C ——>A
Origin State: (3,3,1) = (Mice, Cats, 15-puzzle)
Goal State: (0,0,0)
Operators: Carry (1,1,0) Carry (1,0,0) Carry (0,1,0) Carry (2,0,0) Carry (0,2,0)



Solution:
Carry (0,2,0) -> Goal
Carry (0,1,0) -> Origin
Carry (0,2,0) -> Goal
Carry (0,1,0) -> Origin
Carry (2,0,0) -> Goal
Carry (1,1,0) -> Origin
Carry (2,0,0) -> Goal
Carry (0,1,0) -> Origin
Carry (0,2,0) -> Goal
Carry (1,0,0) -> Origin
Carry (1,1,0) -> Goal

Carry (0,1,0) -> Origin
Carry (0,1,1) -> Goal
The most efficient step = 13
The total steps of ABCA is 21.

C ——>B
Origin State: (2,2,0) = (Mice, Cats, 15-puzzle)
Goal State: (0,0,0)
Operators: Carry (1,1,0) Carry (1,0,0) Carry (0,1,0) Carry (2,0,0) Carry (0,2,0)



Solution:
Carry (2,0,0) -> Goal
Carry (0,1,0) -> Origin
Carry (0,2,0) -> Goal
Carry (0,1,0) -> Origin
Carry (0,2,0) -> Goal
The most efficient step = 5

B ——>A
Origin State: (3,3,1) = (Mice, Cats, 15-puzzle)
Goal State: (0,0,0)
Operators: Carry (1,1,0) Carry (1,0,0) Carry (0,1,0) Carry (2,0,0) Carry (0,2,0)
This is the same as C to A, the solution should be the same. The most efficient step is 13.
For ACBA, the total steps should be 1+5+13+3=22
So, the minimal steps are 21. We should choose the route ABCA.

Question 3

3a
ACTION: **build(FWO)**
PRECOND: obtain (fund), make (plan)
ADD: built(FWO)

ACTION: **launch (material, space)**
PRECOND: assembled (material), on (material, Earth), pre-launch (material)
DELETE: on (material, Earth), pre-launch (material)
ADD: on (material, space)

ACTION: **build (arm)**
PRECOND: on (arm, Earth)
ADD: built (arm)

ACTION: **fold (arm)**
PRECOND: on (arm, Earth), built (arm)
ADD: folded (arm)

ACTION: **pre-launch (arm)**
PRECOND: on (arm, Earth), built (arm)
ADD: pre-launched (arm)

ACTION: **build (mirror)**
PRECOND: on (mirror, Earth)
ADD: built (mirror)

ACTION: **build (laser)**
PRECOND: on (laser, Earth)
ADD: built (laser)

ACTION: **puton (protection, mirror)**
PRECOND: built (mirror)
ADD: have (mirror, protection)

ACTION: **launch (charger, orbit)**
PRECOND: on (charger, Earth), packed (charger), pre-launch (charger)
DELETE: on (charger, Earth), pre-launch (charger)
ADD: in (charger, orbit)

ACTION: **launch (verifier, orbit)**
PRECOND: on (verifier, Earth), packed (verifier), pre-launch (verifier)

DELETE: on (verifier, Earth), pre-launch (verifier)
ADD: in (verifier, orbit)

ACTION: **transport (human, orbit)**
PRECOND: in (human, space station)
DELETE: in (human, space station)
ADD: in (human, orbit)

ACTION: **send (signal, FWO)**
PRECOND: send (request, Earth), program (space station, AGI)
ADD: running (AGI, FWO)

ACTION: **use (AGI)**
PRECOND: running (verifier, FWO)
ADD: run (AGI, FWO)

ACTION: **transport (human, space station)**
PRECOND: in (human, orbit)
DELETE: in (human, orbit)
ADD: in (human, space station)

ACTION: **turnon (laser)**
PRECOND: checked (solar radiation), switched off (laser)
ADD: switched on (laser)

ACTION: **complete (FWO)**
PRECOND: polished (mirror), unfolded (arm), placed (mirror, arm), switched on(laser), in (FWO, orbit), in (human, orbit) or running (AGI, FWO)
DELETE: in (human, orbit)
ADD: completed(FWO)

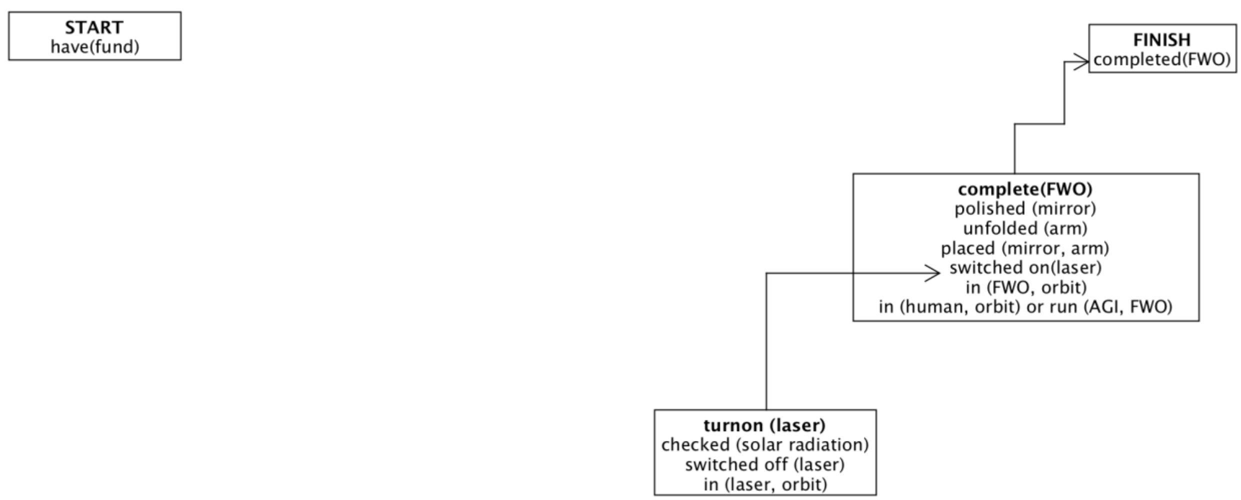
ACTION: **place (mirror, arm)**
PRECOND: built (mirror), unfolded (arm), in (mirror, arm, orbit)
DELETE: polished (mirror)
ADD: placed (mirror, arm)

ACTION: **unfold (arm)**
PRECOND: unpacked (arm)
DELETE: unpacked (arm)
ADD: unfolded (arm)

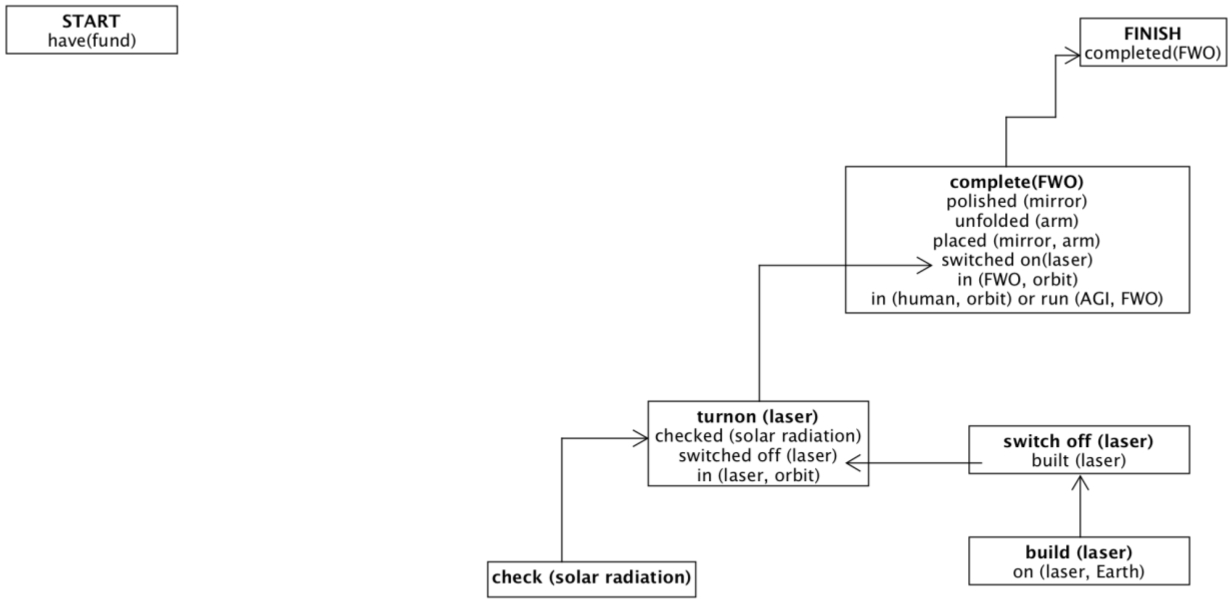
3b

Use the POP algorithm to generate a plan to complete the task. The only action that has COMPLETED as an effect is COMPLETE, so we apply COMPLETE, yielding the following situation:

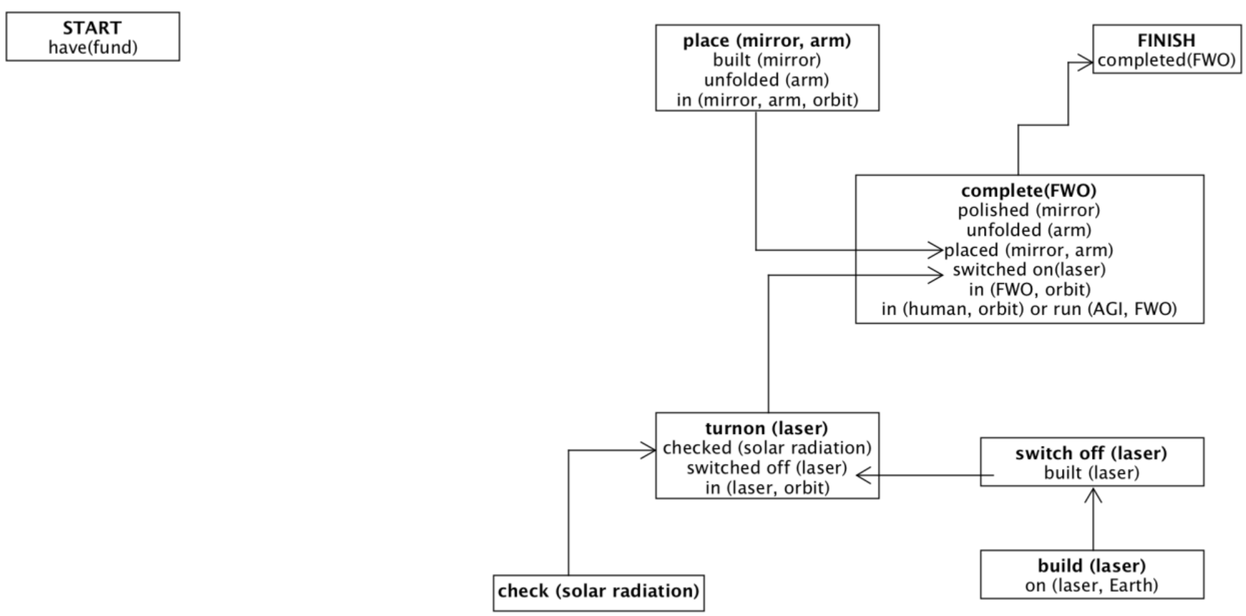
Note: the second IN precondition is optional. The other five unsatisfied preconditions are POLISHED (mirrors), UNFOLDED (arms), PLACED (mirror, arms), SWITCHED ON(laser) and IN (FWO, orbit). The only action that satisfies SWITCHED ON(laser) is TURN ON.



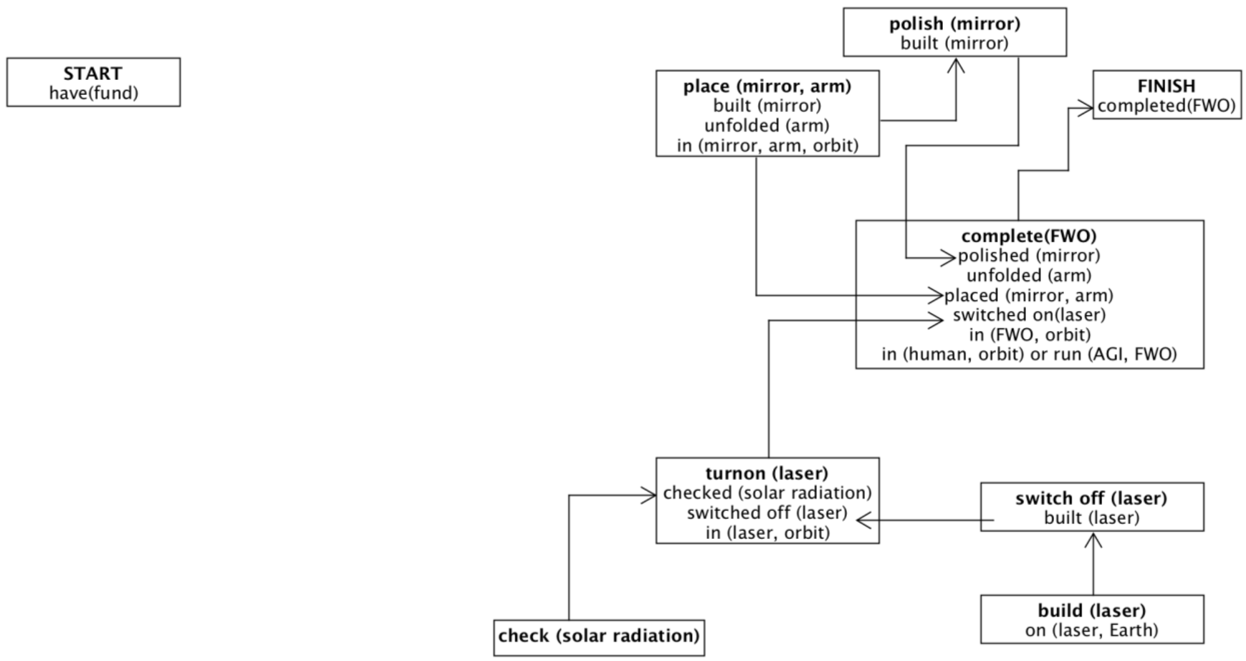
We do not have actions that have CHECKED and SWITCHED OFF as an effect according to the current requirements. Thus, we need to add a new action CHECK and SWITCH OFF.



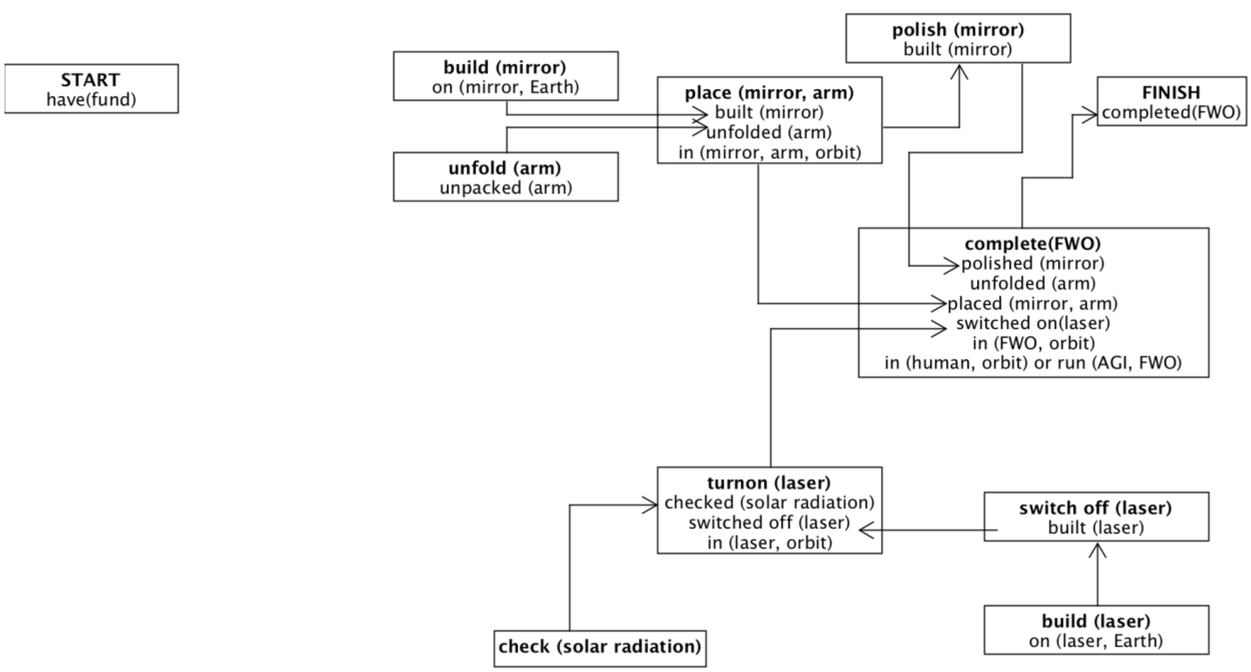
The only action that has PLACED as an effect is PLACE, so we apply PLACE.



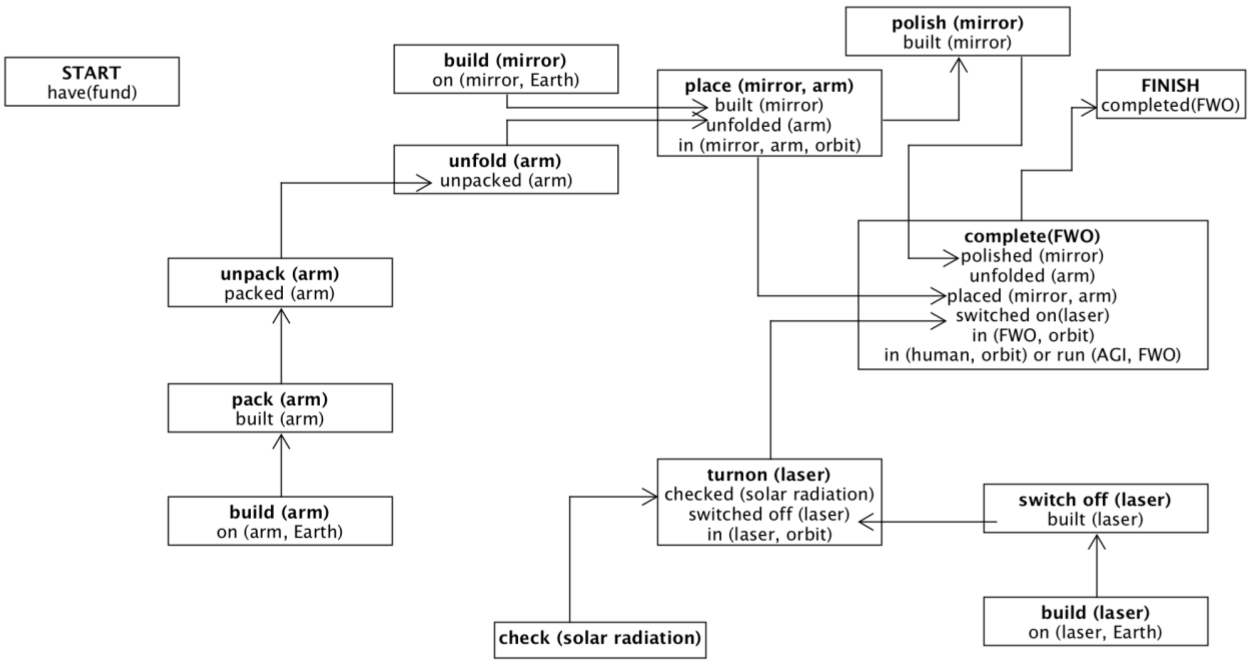
If we polish the mirror before the we place mirrors in arms. It will damage the polish. Thus, we add a POLISH action after PLACE. And it will satisfy the precondition POLISHED in COMPLETE action.



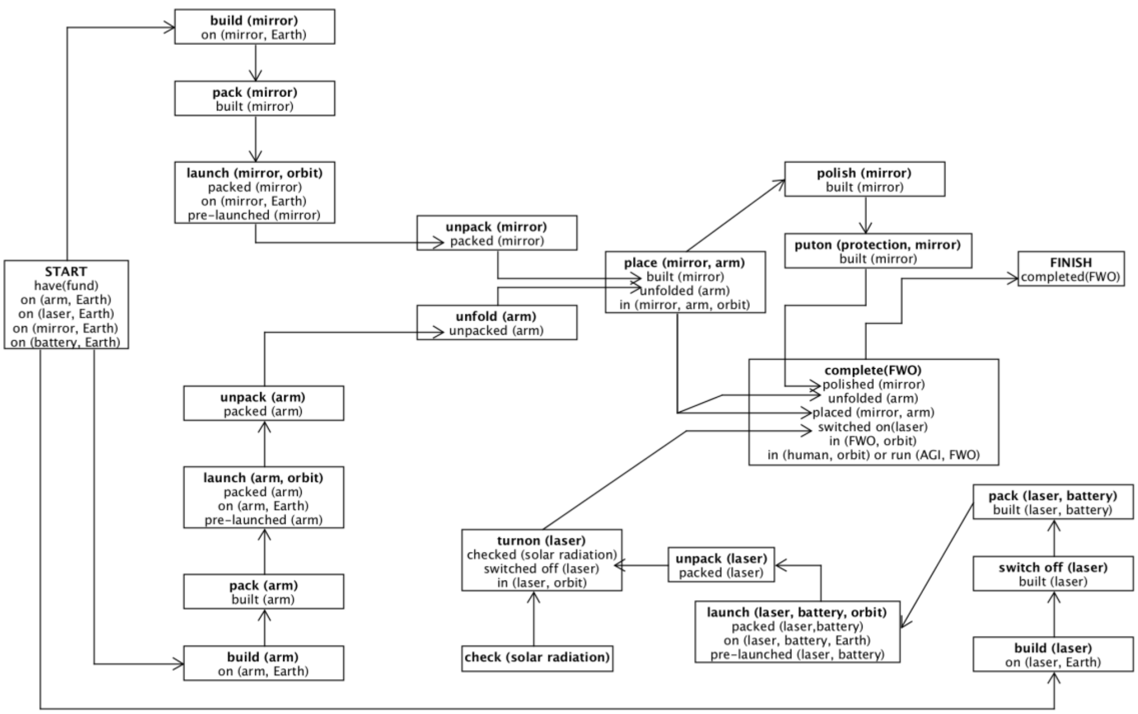
PLACE has two unsatisfied preconditions which are BUILT and UNFOLDED, so we use BUILD and UNFOLD actions.



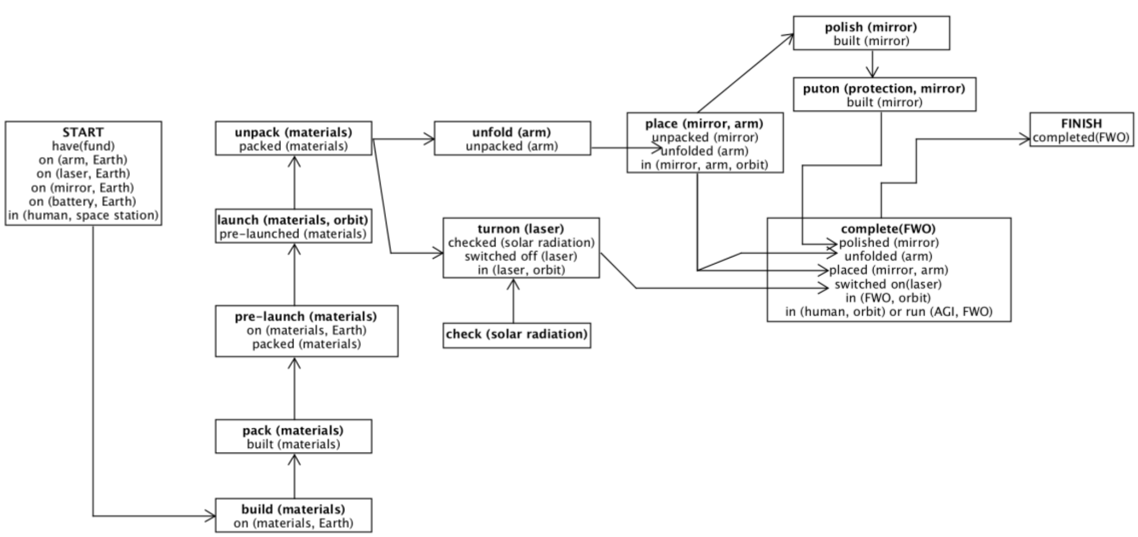
UNFOLD has one unsatisfied precondition which is UNPACKED, so we need to add actions UNPACK and PACK. Moreover, we add two BUILD actions.



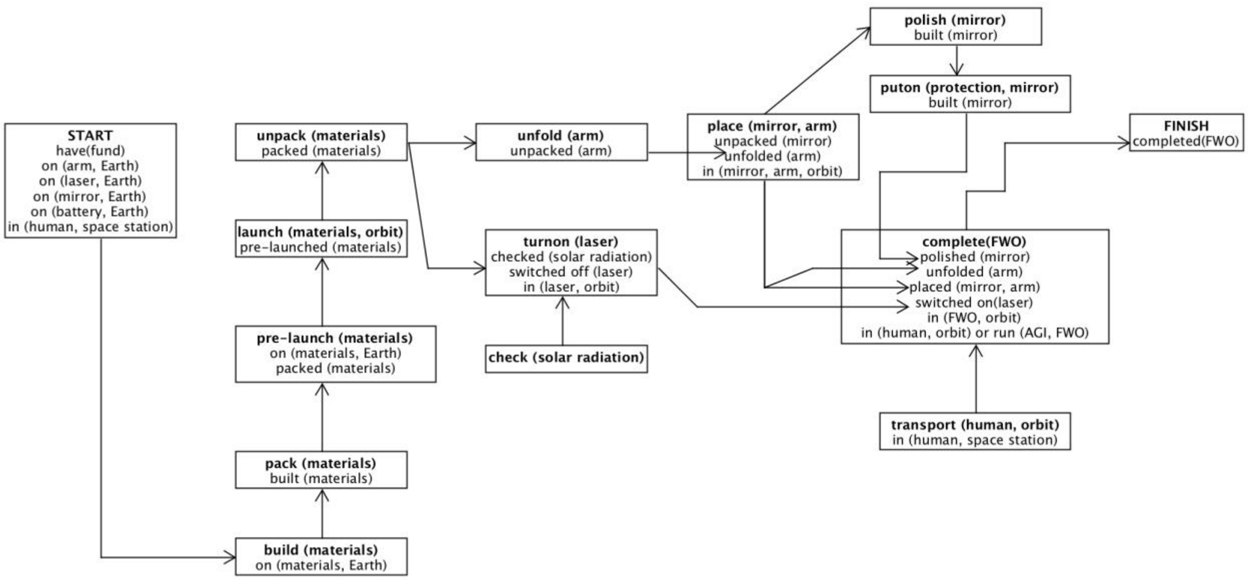
BUILD action can only happen on Earth. Thus, we need to add LAUNCH action. One of the precondition of LAUNCH is PACKED, so that we add LAUNCH after PACK action. The solar panel battery charger is required by laser. So, we need to launch the battery to the orbit too. Then we need to put protection on the mirrors to against both radiation and flare. Furthermore, I assume that all the materials are on Earth at the START state.



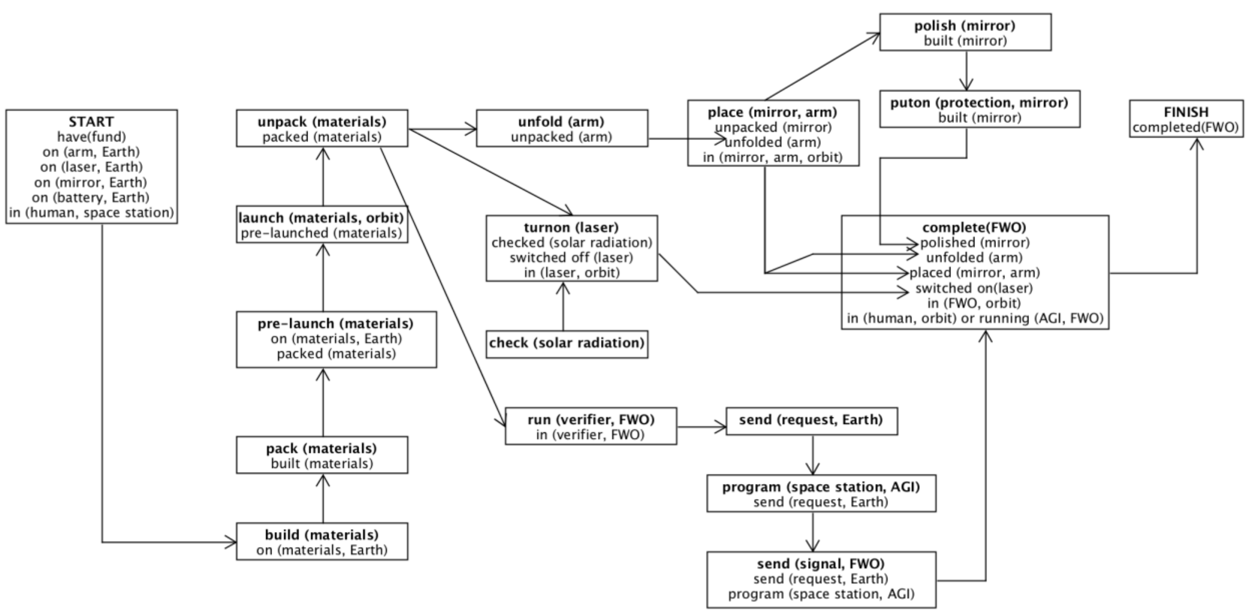
LAUNCH has one unsatisfied precondition which is PRE-LAUNCHED, the only action that has PRE-LAUNCHED as an effect is PRE-LAUNCH, so we apply PRE-LAUNCH. Moreover, we can combine all the mirrors, arms, lasers and chargers into materials. Thus, we can simplify this diagram.



There are two unsatisfied preconditions in COMPLETE action which are IN (human, orbit) and RUN (AGI, FWO). We only need to achieve one of them. In this case, I consider human labour first. They are transported from a nearby space station. I also assume that human labour is in the space station at the START state.



In this second case, I consider to send AGI to FWO. We need to send a software verifier to orbit first. Then send a request signal back to Earth to ask the space station to program the AGI. After then, send the AGI to FWO. We can simply put the software verifier with materials and launch them together. We also need to add RUN action.



Thus, we have two plans that can solve the problem. The first one is sending human labour into FWO:
BUILD (MATERIALS)
PACK (MATERIALS)
PRE-LAUNCH (MATERIALS)
LAUNCH (MATERIALS, ORBIT)
UNPACK (MATERIALS)
CHECK (SOLAR RADIATION)
TURNON (LASER)
UNFOLD (ARM)
PLACE (MIRROR, ARM)
POLISH (MIRROR)
PUTON (PROTECTION, MIRROR)
TRANSPORT (HUMAN, ORBIT)
COMPLETE(FWO)

The second one is programing AGI and send it to FWO:
BUILD (MATERIALS)
PACK (MATERIALS)
PRE-LAUNCH (MATERIALS)
LAUNCH (MATERIALS, ORBIT)
UNPACK (MATERIALS)
CHECK (SOLAR RADIATION)
TURNON (LASER)
UNFOLD (ARM)
PLACE (MIRROR, ARM)
POLISH (MIRROR)
PUTON (PROTECTION, MIRROR)
RUN (VERIFIER, FWO)
SEND (REQUEST, EARTH)
PROGRAM (SPACE STATION, AGI)
SEND (SIGNAL, FWO)
COMPLETE(FWO)

Thus, we should send human labour into FWO instead of programing AGI to complete the task in minimal number of steps.

Question 4

Represent these statements as predicate calculus formulas.

1. All humans are mortal unless they have had their mind uploaded into a computer.
$\forall$x1 $\exists$y [human(x1) $\wedge$ computer(y) $\wedge$ ¬upload_mind(x1, y) $\Rightarrow$ mortal (x1)]

2. All dogs are mortal unless they have had their mind uploaded into a computer.
$\forall$x2 $\exists$y [dog(x2) $\wedge$ computer(y) $\wedge$ ¬upload_mind(x2, y) $\Rightarrow$ mortal (x2)]

3. Mind upload of life form into computer program cannot occur unless either (Star Trek) teleportation can occur or of the following have occurred: driverless cars are safe, and N. Chmait has won the Kurzweil Prize, and the technological singularity has occurred.
$\exists$x $\exists$y [occurred(teleportation) $\vee$ (safe (driveless_car) $\wedge$ won (N.Chmait, Kurzweil Prize) $\wedge$ occurred (Technological Singularity)) $\Rightarrow$ life_form(x) $\wedge$ computer(y) $\wedge$ Uploads(x, y)]

4. The technological singularity has not occurred.
¬occurred (Technological Singularity)

5. N. Chmait won the Kurzweil Prize in 2017.
won (N.Chmait, Kurzweil Prize)

6. Socrates lives in a household whose only residents are humans and dogs.
$\exists$l $\exists$ x1 $\exists$ x2[human(x1) $\wedge$ dog(x2) $\wedge$ house(l) $\Rightarrow$ lives(Socrates, l) $\wedge$ lives(j, l) $\vee$ lives(k, l)]

7. Humans and dogs are both life forms.
$\forall$x1 $\forall$x2[human(x1) $\Rightarrow$ life_form(x1) $\wedge$ dog(x2) $\Rightarrow$ life_form(x2)]

8. Teleportation requires quantum mechanical tunnelling and the creation of two strong plasma fields.
$\exists$a $\exists$b1 $\exists$b2 [quantum mechanical tunnelling(a) $\wedge$ plasma(b1) $\wedge$plasma (b2) $\Rightarrow$ occurred(Teleportation)]

9. When quantum mechanical tunnelling occurs, it is possible to have at most one strong plasma field.
$\exists$a ¬ $\exists$b1¬ $\exists$b2 [quantum mechanical tunnelling(a) $\Rightarrow$ plasma(b1) $\wedge$plasma(b2)]

Convert these statements to clauses.

convert statement 1 to clauses:
$\forall$x1 $\exists$y [human(x1) $\wedge$ computer(y) $\wedge$ ¬upload_mind(x1, y) $\Rightarrow$ mortal (x1)]
$\forall$x1 [human(x1) $\wedge$ computer(Y) $\wedge$ ¬upload_mind(x1, Y) $\Rightarrow$ mortal (x1)]

¬human(x1) ∨ ¬computer(Y) ∨ upload_mind(x1, Y) ∨ mortal (x1)

convert statement 2 to clauses:
∀x2∃y [dog(x2) ∧ computer(y) ∧ ¬upload_mind(x2, y) ⇒ mortal (x2)]
¬dog(x2) ∨ ¬computer(Y) ∨ upload_mind(x2, Y) ∨ mortal (x2)

convert statement 3 to clauses:
∃x∃y [occurred(teleportation) ∨ (safe (driveless_car) ∧ won (N.Chmait, Kurzweil Prize) ∧ occurred (Technological Singularity)) ⇒ life_form(x) ∧ computer(y) ∧ Uploads(x, y)]
¬(occurred(teleportation) ∨ (safe (driveless_car) ∧ won (N.Chmait, Kurzweil Prize) ∧ occurred (Technological Singularity))) ∨ (life_form(x) ∧ computer(y) ∧ Uploads(x, y))
occurred(teleportation) ∨ (safe (driveless_car) ∧ won (N.Chmait, Kurzweil Prize) ∧ occurred (Technological Singularity)) ∧ ¬(life_form(x) ∧ computer(y) ∧ Uploads(x, y))
occurred(teleportation) ∨ (safe (driveless_car) ∧ won (N.Chmait, Kurzweil Prize) ∧ occurred (Technological Singularity)) ∧ ¬life_form(x) ∨¬computer(y) ∨¬Uploads(x, y)
occurred(teleportation) ∧ (safe (driveless_car) ∨ occurred(teleportation) ∧ won (N.Chmait, Kurzweil Prize) ∨ occurred(teleportation) ∧ occurred (Technological Singularity)
∧ ¬life_form(x) ∨¬computer(y) ∨¬Uploads(x, y)
3a. ccurred(teleportation) ∧ safe (driveless_car)
3b. occurred(teleportation) ∧ won (N.Chmait, Kurzweil Prize)
3c. occurred(teleportation) ∧ occurred (Technological Singularity)
3d. ¬life_form(x) ∨¬computer(y) ∨¬Uploads(x, y)

convert statement 6 to clauses:
∃l∃ x1∃ x2[human(x1) ∧ dog(x2) ∧ house(l) ⇒ lives(Socrates, l) ∧ lives(j, l) ∨ lives(k, l)]
∃l∃ x1∃ x2[¬(human(x1) ∧ dog(x2) ∧ house(l)) ∨lives(Socrates, l) ∧ lives(j, l) ∨ lives(k, l)]
¬(human(x1) ∧ dog(x2) ∧ house(l)) ∨lives(Socrates, l) ∧ lives(j, l) ∨ lives(k, l)
human(x1) ∧ dog(x2) ∧ house(l) ∧¬lives(Socrates, l) ∨ ¬lives(j, l) ∨ ¬lives(k, l)

convert statement 7 to clauses:
∀x1∀x2[human(x1) ⇒ life_form(x1) ∧ dog(x2) ⇒ life_form(x2)]
7a. ¬human(x1) ∨ life_form(x1)
7b. ¬ dog(x2) ∨ life_form(x2)

convert statement 8 to clauses:
∃a∃b1∃b2 [quantum mechanical tunnelling(a) ∧ plasma(b1) ∧plasma (b2) ⇒ occurred(Teleportation)]
∃a∃b1∃b2∃c [¬(quantum mechanical tunnelling(a) ∧ plasma(b1) ∧plasma (b2)) ∨ occurred(Teleportation)]
∃a∃b1∃b2∃c [¬quantum mechanical tunnelling(a) ∨ ¬plasma(b1) ∨¬plasma (b2) ∨ occurred(Teleportation)]
¬quantum mechanical tunnelling(A) ∨ ¬plasma(B1) ∨¬plasma (B2) ∨occurred(Teleportation)

convert statement 9 to clauses:
$\exists a \neg \exists b1 \neg \exists b2$ [quantum mechanical tunnelling(a) $\Rightarrow$ plasma(b1) $\wedge$ plasma(b2)]
$\exists a \neg \exists b1 \neg \exists b2$ [¬quantum mechanical tunnelling(a) $\vee$ plasma(b1) $\wedge$ plasma(b2)]
$\neg \exists b1 \neg \exists b2$ [¬quantum mechanical tunnelling(A) $\vee$ plasma(b1) $\wedge$ plasma(b2)]
$\forall b1 \forall b2$[¬quantum mechanical tunnelling(A) $\vee$ ¬plasma(b1) $\vee$ ¬plasma(b2)]
¬quantum mechanical tunnelling(A) $\vee$ ¬plasma(B1) $\vee$ ¬plasma(B2)

Use resolution to prove that Socrates is mortal

Goal: mortal(Socrates)
Negated goal: 10. ¬mortal(Socrates)

10 and 1:
10. ¬mortal(Socrates)
1.¬human(x1) $\vee$ ¬computer(Y) $\vee$ upload_mind(x1, Y) $\vee$ mortal (x1)
mgu: {x1/Socrates}
resolvent: 11. ¬human(Socrates) $\vee$ ¬computer(Y) $\vee$ upload_mind(Socrates, Y)

11 and 3d:
11. ¬human(Socrates) $\vee$ ¬computer(Y) $\vee$ upload_mind(Socrates, Y)
3d. ¬life_form(x) $\vee$¬computer(y) $\vee$¬Uploads(x, y)
mgu: {x/Socrates}
resolvent: 12. ¬human(Socrates) $\vee$ ¬computer(Y) $\vee$¬life_form(Socrates)

12 and 7b:
12. ¬human(Socrates) $\vee$ ¬computer(Y) $\vee$¬life_form(Socrates)
7b. ¬human(x1) $\vee$ life_form(x1)
mgu: {x1/Socrates}
resolvent:13. ¬human(Socrates) $\vee$ ¬computer(Y)

Question 5

5a

P(X|¬Test) = 1 – 90% = 10%

5b

P(Test|X) = P(X|Test) *P(X)/(P(X|Test) *P(X) + P(X|¬Test) *P(¬X))

= 0.9*0.01/ (0.9*0.01 + 0.1*0.99)

= 0.083

= 8.3%

The probability of a patient who tests positive has condition X is 8.3%

5c

P(Test|X) = P(X|Test) *P(X)/(P(X|Test) *P(X) + P(X|¬Test) *P(¬X))

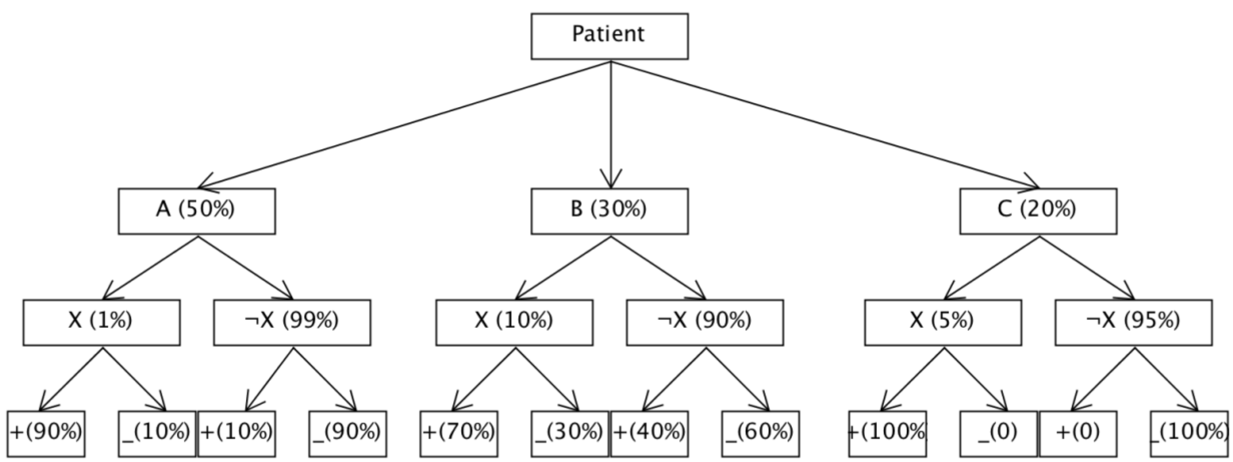
0.2 = 0.6P(X)/ (0.6P(X) + 0.1(1-P(X)))

P(X) = 0.04

= 4%

4% of people in town D have condition X

5d



5d (i)

P(+|A) = P(A|+) / P (+)

= 0.5* (0.01 * 0.9 + 0.99 * 0.1) / (0.5* (0.01 * 0.9 + 0.99 * 0.1) + 0.3* (0.1 * 0.7 + 0.9 * 0.4) + 0.2* (0.05 * 1))

= 0.054 / 0.193

= 28%

P(+|B) = P(B|+) / P (+)

= 0.3* (0.1 * 0.7 + 0.9 * 0.4) / (0.5* (0.01 * 0.9 + 0.99 * 0.1) + 0.3* (0.1 * 0.7 + 0.9 * 0.4) + 0.2* (0.05 * 1))

= 0.129/ 0.193

= 66.8%

P(+|C) = P(C|+) / P (+)

= 0.3* (0.1 * 0.7 + 0.9 * 0.4) / (0.5* (0.01 * 0.9 + 0.99 * 0.1) + 0.3* (0.1 * 0.7 + 0.9 * 0.4) + 0.2* (0.05 * 1))

= 0.01/ 0.193

= 5.2%

5d (ii)

P(¬X|A) = P(A|¬X)/ P(¬X)

= 0.5*0.99 / (0.5*0.99 + 0.3*0.9 + 0.2*0.95)

= 0.495 / 0.955

= 51.8%

P(¬X|B) = P(B|¬X)/ P(¬X)

= 0.3*0.9 / (0.5*0.99 + 0.3*0.9 + 0.2*0.95)

= 0.27 / 0.955

= 28.3%

P(¬X|C) = P(C|¬X)/ P(¬X)

= 0.2*0.95 / (0.5*0.99 + 0.3*0.9 + 0.2*0.95)

= 0.19 / 0.955

= 19.9%

5d (iii)

P(¬X|Test) = P(Test|¬X)/ P(¬X)

= (0.5*0.99*0.1 + 0.3*0.9*0.4 + 0)/ (0.5*0.99 + 0.3*0.9 + 0.2*0.95)

= 0.1575/ 0.955

= 16.5%