**Artificial Intelligence & Machine Learning – September 5, 2024**

**Part 1 - Artificial Intelligence**

*(Time to complete the test: 2:30 hours)*

A spaceship can travel among planets. Every planet hosts a fuel station where the spaceship can be fuelled. The spaceship's fuel tank has a limited capacity, and the tank level can be empty, half or full. When the spaceship moves from one planet to another, the amount of fuel consumed reduces from full to half and from half to empty.

Assume the scenario is modelled as follows:

*non-Fluents*:

- $Planet(x)$ denoting that $x$ is a planet;

*Fluents*:

- $Full()$ denoting that the tank is full;
- $Half()$ denoting that the tank is half;
- $Empty()$ denoting that the tank is empty;
- $On(x)$ denoting that the spaceship is on planet $x$.

*Actions*:

- $move(x, y)$, which allows the spaceship to travel from planet $x$ to planet $y$.
  The action can be done only if:
  - the spaceship is on planet $x$;
  - $y$ is a planet;
  - $x$ and $y$ are distinct;
  - the fuel level is not empty.

  The effect is that:
  - the spaceship is in $y$ and not in $x$ anymore, and the fuel level changes from full to half and from half to empty.
- $refuel()$, which allows the spaceship to fill up the fuel tank. The action can be done only if the spaceship is on some planet. The effect is that the fuel level becomes full (and not any other level).

*Initial situation*:
There are four planets: Earth, Mars, Jupiter, Saturn. The spaceship is initially on Earth with an empty tank.

**Exercise 1.**

1. Formalize the above scenario as a Basic Action Theory.
2. Using regression, check whether the action sequence

$$\varrho_1 = refuel(); move(Earth, Jupiter); move(Jupiter, Earth);$$

   is executable in $S_0$.
3. Using regression, check whether $\varrho_1$ leads to a situation where the tank is empty.

**Exercise 2.**

1. Considering goal $\gamma = On(Saturn) \land Empty()$, formalize the above scenario as a PDDL domain and a PDDL problem files;
2. Draw the corresponding (entire) transition system;
3. Solve planning for achieving $\gamma$ using uninformed forward depth-first search, reporting the steps of the forward search computation and showing, in particular, the evolution of the open set (stack). Report the returned plan.

**Exercise 3.**

1. Using the tableau method, check whether the following holds:

$$\forall x.P(x) \supset \exists z.Q(z) \models \forall z.P(z) \supset \exists x \exists y P(x) \wedge Q(y)$$

   If this is not the case, show a counter-model obtained from the tableau.

**Exercise 1.**

1. Formalize the above scenario as a Basic Action Theory.
2. Using regression, check whether the action sequence

$$\varrho_1 = refuel(); move(Earth, Jupiter); move(Jupiter, Earth);$$

is executable in $S_0$.

3. Using regression, check whether $\varrho_1$ leads to a situation where the tank is empty.

1) PRECONDITIONS AXIOMS

$$Poss(move(x,y),s) \equiv On(x,s) \wedge PLANET(y) \wedge x \neq y \wedge \neg EMPTY(s)$$
$$Poss(REFUEL(),s) \equiv \exists x.\, On(x,s)$$

SUCCESSOR STATE AXIOMS

EFFECT AXIOMS

$$a = move(x,y) \supset \neg On(x, DO(a,s)) \wedge On(y, DO(a,s)) \wedge (FULL(s) \subset \neg FULL(DO(a,s)) \wedge HALF(DO(a,s))) \wedge (HALF(S) \subset \neg HALF(DO(a,s)) \wedge EMPTY(DO(a,s)))$$

$$a = REFUEL() \supset FULL(DO(a,s)) \wedge \neg HALF(DO(a,s)) \wedge \neg EMPTY(DO(a,s))$$

NORMALIZE

$$\exists y\ a = move(x,y) \supset \neg On(x, DO(a,s))$$

$$\exists x.\ a = move(x,y) \supset ON(y, DO(a,s))$$

$$\exists x,y.(a = move(x,y) \wedge FULL(s)) \supset \neg FULL(DO(a,s))$$

$$\exists x,y.(a = move(x,y) \wedge FULL(s)) \supset HALF(DO(a,s))$$

$$\exists x,y.(a = move(x,y) \wedge HALF(S)) \supset \neg HALF(DO(a,s))$$

$$\exists x,y.(a = move(x,y) \wedge HALF(S)) \supset EMPTY(DO(a,s))$$

$$a = REFUEL() \supset FULL(DO(a,s))$$

$$a = REFUEL() \supset \neg HALF(DO(a,s))$$

$$a = REFUEL() \supset \neg EMPTY(DO(a,s))$$

## EXPLANATION CLOSURE

$$FULL(DO(a,s)) \equiv a = REFUEL() \lor (FULL(S) \land \neg \exists x,y.\, a = move(x,y))$$

$$HALF(DO(a,s)) \equiv \exists x,y.(a = move(x,y) \land FULL(s)) \lor (HALF(s) \land \neg((\exists x,y.\, a = move(x,y)) \lor a = REFUEL())$$

$$EMPTY(DO(a,s)) \equiv \exists x,y.(a = move(x,y) \land HALF(s)) \lor (EMPTY(s) \land \neg a = REFUEL())$$

$$ON(x, DO(a,s)) \equiv (\exists y.\, a = move(y,x)) \lor (ON(x,s) \land \neg \exists y.\, a = move(x,y))$$

## INITIAL SITUATION

$PLANET(x) \equiv (x = EARTH) \lor (x = MARS) \lor (x = JUPITER) \lor (x = SATURN)$
$ON(x,S_0) \equiv x = EARTH$
$EMPTY(S_0)$
$\neg FULL(S_0)$
$\neg HALF(S_0)$

2) $\varrho_1 = refuel(); move(Earth, Jupiter); move(Jupiter, Earth);$ EXECUTABLE IN $S_0$

$S_1 = DO(REFUEL(), S_0)$
$S_2 = DO(MOVE(EARTH, JUPITER), S_1)$
$S_3 = DO(MOVE(JUPITER, EARTH), S_2)$

WE HAVE TO CHECK IF

REFUEL() IS EXECUTABLE IN $S_0$
MOVE(EARTH, JUPITER) IS EXECUTABLE IN $S_1$
MOVE(JUPITER, EARTH) IS EXECUTABLE IN $S_2$

1. $D_0 \cup D_{una} \models \mathcal{R}[Poss(REFUEL(), S_0)]$
2. $D_0 \cup D_{una} \models \mathcal{R}[Poss(MOVE(EARTH, JUPITER), S_1)]$
3. $D_0 \cup D_{una} \models \mathcal{R}[Poss(MOVE(JUPITER, EARTH), S_2)]$

LET'S REGRESS EACH FORMULA

1. $\mathcal{R}[Poss(REFUEL(), S_0)] = \mathcal{R}[\exists x.\, ON(x, S_0)] = \exists x.\, ON(x, s)$

TRUE FOR $x = EARTH$ → REFUEL IS EXECUTABLE IN $S_0$ ✓

2. $\mathcal{R}[Poss(MOVE(EARTH, JUPITER), S_1)] =$
$= \mathcal{R}[ON(EARTH, S_1) \wedge PLANET(JUPITER) \wedge EARTH \neq JUPITER \wedge \neg EMPTY(S_1)]$
$= \mathcal{R}[ON(EARTH, S_1)] \wedge \mathcal{R}[\neg EMPTY(S_1)]$

$\mathcal{R}[ON(EARTH, S_1)] = \mathcal{R}[ON(EARTH, DO(REFUEL(), S_0))] =$
$= \mathcal{R}[(\exists y.\ REFUEL() = move(y, EARTH)) \vee (ON(EARTH, S_0) \wedge$
$\neg \exists y. REFUEL() = move(EARTH, y)] =$
$= \mathcal{R}[TRUE] = TRUE$

$\mathcal{R}[\neg EMPTY(S_1)] = \neg \mathcal{R}[EMPTY(DO(REFUEL(), S_0)] =$
$= \neg \mathcal{R}[(\exists x, y.\ REFUEL() = move(x, y) \wedge HALF(S_0)) \vee$
$(EMPTY(S_0) \wedge \neg REFUEL() = REFUEL())] =$
$= \neg \mathcal{R}[FALSE] = TRUE$

$\mathcal{R}[Poss(MOVE(EARTH, JUPITER), S_1)] =$
$= \mathcal{R}[ON(EARTH, S_1)] \wedge \mathcal{R}[\neg EMPTY(S_1)] = TRUE$

MOVE (EARTH, JUPITER) IS EXECUTABLE IN $S_1$ ✓

3. $\mathcal{R}[Poss(MOVE(JUPITER, EARTH), S_2)] =$
$= \mathcal{R}[ON(JUPITER, S_2) \wedge PLANET(EARTH) \wedge JUPITER \neq EARTH \wedge \neg EMPTY(S_2)]$
$= \mathcal{R}[ON(JUPITER, S_2)] \wedge \neg \mathcal{R}[EMPTY(S_2)]$

$\mathcal{R}[ON(JUPITER, S_2)] = \mathcal{R}[ON(JUPITER, DO(MOVE(EARTH, JUPITER), S_1)] =$
$= \mathcal{R}[(\exists y.\ MOVE(EARTH, JUPITER) = move(y, JUPITER)) \vee$
$(ON(JUPITER, S_1) \wedge \neg \exists y.\ MOVE(EARTH, JUPITER) = move(EARTH, y))] =$
$= \mathcal{R}[TRUE] = TRUE$

$\neg \mathcal{R}[EMPTY(S_2)] = \neg \mathcal{R}[EMPTY(DO(MOVE(EARTH, JUPITER), S_1))] =$
$= \neg \mathcal{R}[(\exists x, y.\ MOVE(EARTH, JUPITER) = move(x, y) \wedge HALF(S_1)) \vee$
$(EMPTY(S_1) \wedge \neg MOVE(EARTH, JUPITER) = REFUEL())] =$
$= \neg \mathcal{R}[FALSE] = TRUE$

$\mathcal{R}[Poss(MOVE(JUPITER, EARTH), S_2)] =$
$= \mathcal{R}[ON(JUPITER, S_2)] \wedge \neg \mathcal{R}[EMPTY(S_2)] = TRUE$

MOVE (JUPITER, EARTH) IS EXECUTABLE IN $S_2$ ✓

SO $P_1$ IS EXECUTABLE IN $S_0$ ✓

3) $\varrho_1 = refuel(); move(Earth, Jupiter); move(Jupiter, Earth);$ THE TANK IS EMPTY

WE HAVE TO CHECK $\quad D_0 \cup D_{una} \models R[EMPTY(S_3)]$

$S_1 = DO(REFUEL(), S_0)$
$S_2 = DO(MOVE(EARTH, JUPITER), S_1)$
$S_3 = DO(MOVE(JUPITER, EARTH), S_2)$

$R[EMPTY(S_3)] = R[EMPTY(DO(MOVE(JUPITER, EARTH), S_2))] =$
$= R[(\exists x,y.\ MOVE(JUPITER, EARTH) = MOVE(x,y) \wedge HALF(S_2)) \vee$
$(EMPTY(S_2) \wedge \neg MOVE(JUPITER, EARTH) = REFUEL())] =$
$= R[HALF(S_2)] \vee R[EMPTY(S_2)]$

$R[HALF(S_2)] = R[HALF(DO(MOVE(EARTH, JUPITER), S_1)] =$
$= R[(\exists x,y.\ MOVE(EARTH, JUPITER) = MOVE(x,y) \wedge FULL(S_1)) \vee (HALF(S_1) \wedge$
$\neg(\exists x,y.\ MOVE(EARTH, JUPITER) = MOVE(x,y) \vee MOVE(EARTH, JUPITER) = REFUEL())] =$
$= R[FULL(S_1)] =$
$= R[FULL(DO(REFUEL(), S_0))] =$
$= R[REFUEL() = REFUEL() \vee (FULL(S_0) \wedge \neg \exists x,y.\ REFUEL() = move(x,y))] =$
$= R[TRUE] = TRUE$

$R[EMPTY(S_2)] = R[EMPTY(DO(MOVE(EARTH, JUPITER), S_1))] =$
$= R[(\exists x,y.\ MOVE(EARTH, JUPITER) = MOVE(x,y) \wedge HALF(S_1)) \vee$
$(EMPTY(S_1) \wedge \neg MOVE(EARTH, JUPITER) = REFUEL())] =$
$= R[HALF(S_1)] \vee R[EMPTY(S_1)] =$

$R[HALF(S_1)] = R[HALF(DO(REFUEL(), S_0))] =$
$= R[(\exists x,y.\ REFUEL() = MOVE(x,y) \wedge FULL(S_0)) \vee$
$(HALF(S_0) \wedge \neg(\exists x,y.\ REFUEL() = MOVE(x,y) \vee REFUEL() = REFUEL())] =$
$= R[FALSE] = FALSE$

$R[EMPTY(S_1)] = R[EMPTY(DO(REFUEL(), S_0))] =$
$= R[(\exists x,y.\ REFUEL() = MOVE(x,y) \wedge HALF(S_0)) \vee$
$(EMPTY(S_0) \wedge \neg REFUEL() = REFUEL())] =$
$= R[FALSE] = FALSE$

$R[EMPTY(S_2)] = R[HALF(S_1)] \vee R[EMPTY(S_1)] = FALSE$

$R[EMPTY(S_3)] = R[HALF(S_2)] \vee R[EMPTY(S_2)] = TRUE$

$\rho_1$ LEADS TO A SITUATION WHERE THE TANK IS EMPTY ✓

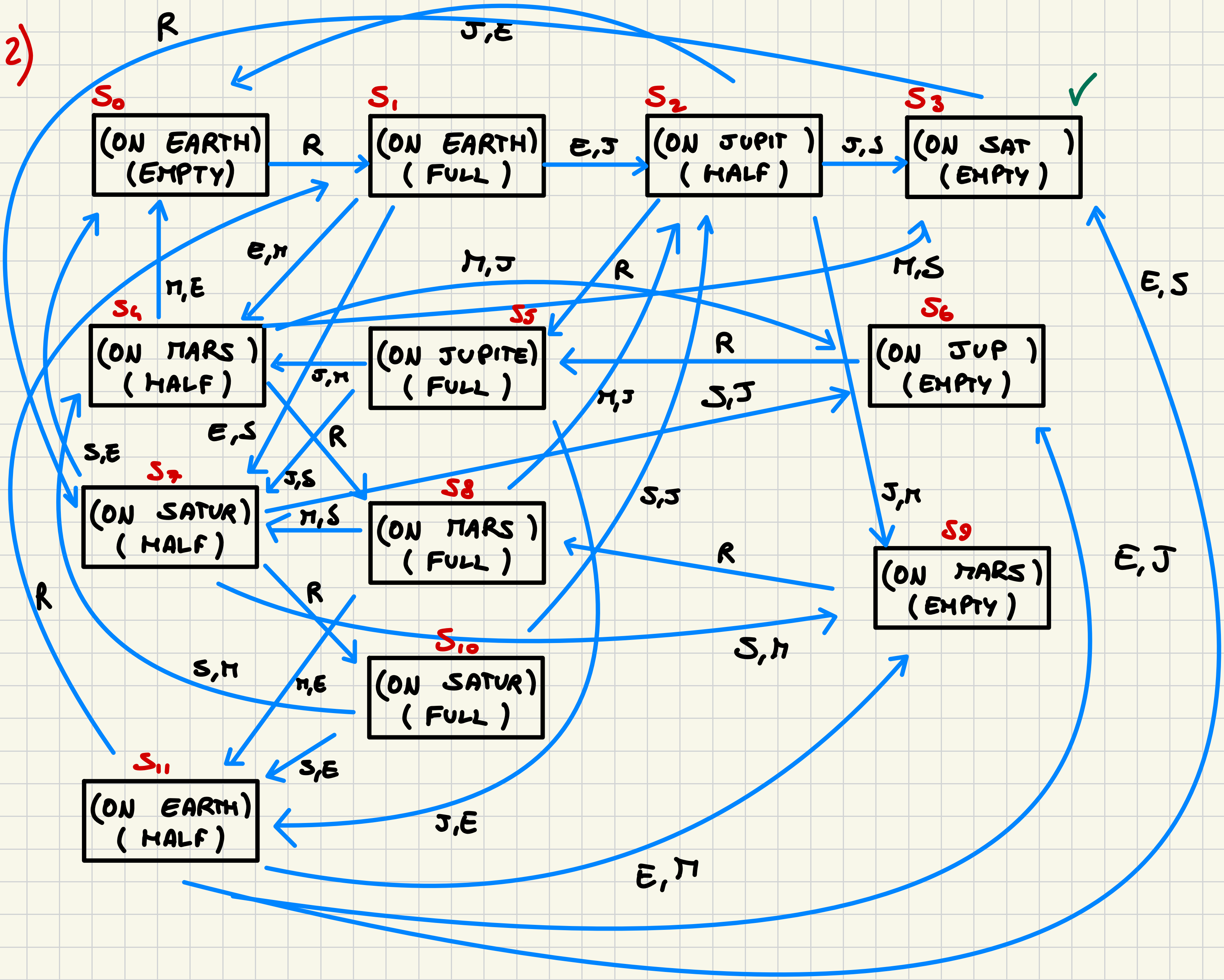
**Exercise 2.**

1. Considering goal $\gamma = On(Saturn) \wedge Empty()$, formalize the above scenario as a PDDL domain and a PDDL problem files;
2. Draw the corresponding (entire) transition system;
3. Solve planning for achieving $\gamma$ using uninformed forward depth-first search, reporting the steps of the forward search computation and showing, in particular, the evolution of the open set (stack). Report the returned plan.

1)
```
(DEFINE (DOMAIN SPACE_DOMAIN)
  (:REQUIREMENTS :ADL)
  (:TYPES PLANET)
  (: PREDICATES
       (HALF)
       (FULL)
       (EMPTY)
       (PLANET ?x -PLANET)
       (ON ?x - PLANET)
  )
  (:ACTION MOVE
     :PARAMETERS ( ?x ?y -PLANET)
     :PRECONDITIONS (AND
                       (ON ?x)(PLANET ?y)
                       (NOT (= ?x ?y))(NOT (EMPTY))
                    )
     :EFFECT (AND
                (NOT (ON ?x))
                (ON ?y)
                (WHEN (FULL)(HALF))
                (WHEN (HALF)(EMPTY))
             )
  ); END OF MOVE
  (:ACTION REFUEL
     :PARAMETERS ()
     :PRECONDITIONS (EXISTS (?x - PLANET) (ON ?x))
     : EFFECT (AND
                 (FULL) (NOT (HALF))(NOT (EMPTY))
              )
  ); END OF REFUEL
); END OF DEFAIN DOMAIN

(DEFINE (PROBLEM SPACE_PROBLEM)(: DOMAIN SPACE_DOMAIN)
  (: OBJECTS  EARTH MARS  JUPITER SATURN - PLANET)
  (: INIT  (ON EARTH)(EMPTY))
  (: GOAL ( AND
            (ON SATURN)(EMPTY)
          )
  )
); END OF DEFINE PROBLEM
```

2)


S0
(ON EARTH)
(EMPTY)
S1
(ON EARTH)
( FULL )
S2
(ON JUPIT )
( HALF )
S3
(ON SAT )
( EMPTY )
S4
(ON MARS )
( HALF )
S5
(ON JUPITE)
( FULL )
S6
(ON JUP )
( EMPTY )
S7
(ON SATUR)
( HALF )
S8
(ON MARS )
( FULL )
S9
(ON MARS)
( EMPTY )
S10
(ON SATUR)
( FULL )
S11
(ON EARTH)
( HALF )
R
J,E
E,J
J,S
E,M
M,E
M,J
M,S
E,S
J,M
S,J
S,E
J,S
M,S
S,M
M,E
S,E
J,E
E,M
E,J

3) $\gamma = On(Saturn) \wedge Empty()$

0. $\pi$ = [($S_0$, EMPTY)]
   m : {$S_0$}

1. (STATE, PLAN) = $\pi$. POP()      ($S_0$, EMPTY)
   m. ADD ($S_1$)
   $\pi$. PUSH ($S_1$, REFUEL())

   $\pi$ = [($S_1$, REFUEL())]      m: {$S_0$, $S_1$}

2. (STATE, PLAN) = $\pi$. POP()      ($S_1$, REFUEL())
   m. ADD ($S_4$)
   $\pi$. PUSH ($S_4$, REFUEL() MOVE (EARTH, MARS))
   m. ADD ($S_7$)
   $\pi$. PUSH ($S_7$, REFUEL() MOVE (EARTH, SATURN))
   m. ADD ($S_2$)
   $\pi$. PUSH ($S_2$, REFUEL() MOVE (EARTH, JUPITER))

   $\pi$ = [ ($S_2$, REFUEL() MOVE (EARTH, JUPITER)),
   ($S_7$, REFUEL() MOVE (EARTH, SATURN)),
   ($S_4$, REFUEL() MOVE (EARTH, MARS))],      m: {$S_0$, $S_1$, $S_2$, $S_4$, $S_7$}

3. (STATE, PLAN) = $\pi$. POP()      ($S_2$, REFUEL() MOVE (EARTH, JUPITER))
   m. ADD ($S_5$)
   $\pi$. PUSH ($S_5$, REFUEL() MOVE (EARTH, MARS) REFUEL())
   m. ADD ($S_9$)
   $\pi$. PUSH ($S_9$, REFUEL() MOVE (EARTH, SATURN) MOVE (JUPITER, MARS))
   m. ADD ($S_3$)
   $\pi$. PUSH ($S_3$, REFUEL() MOVE (EARTH, JUPITER) MOVE (JUPITER, SATURN))

   $\pi$ = [($S_3$, REFUEL() MOVE (EARTH, JUPITER) MOVE (JUPITER, SATURN)),
   ($S_9$, REFUEL() MOVE (EARTH, SATURN) MOVE (JUPITER, MARS)),
   ($S_5$, REFUEL() MOVE (EARTH, MARS) REFUEL()),
   ($S_7$, REFUEL() MOVE (EARTH, SATURN)),
   ($S_4$, REFUEL() MOVE (EARTH, MARS))],      m: {$S_0$, $S_1$, $S_2$, $S_3$, $S_4$, $S_5$, $S_7$, $S_9$}

4. (STATE, PLAN) = $\pi$. POP()

   ($S_3$, REFUEL() MOVE (EARTH, JUPITER) MOVE (JUPITER, SATURN))

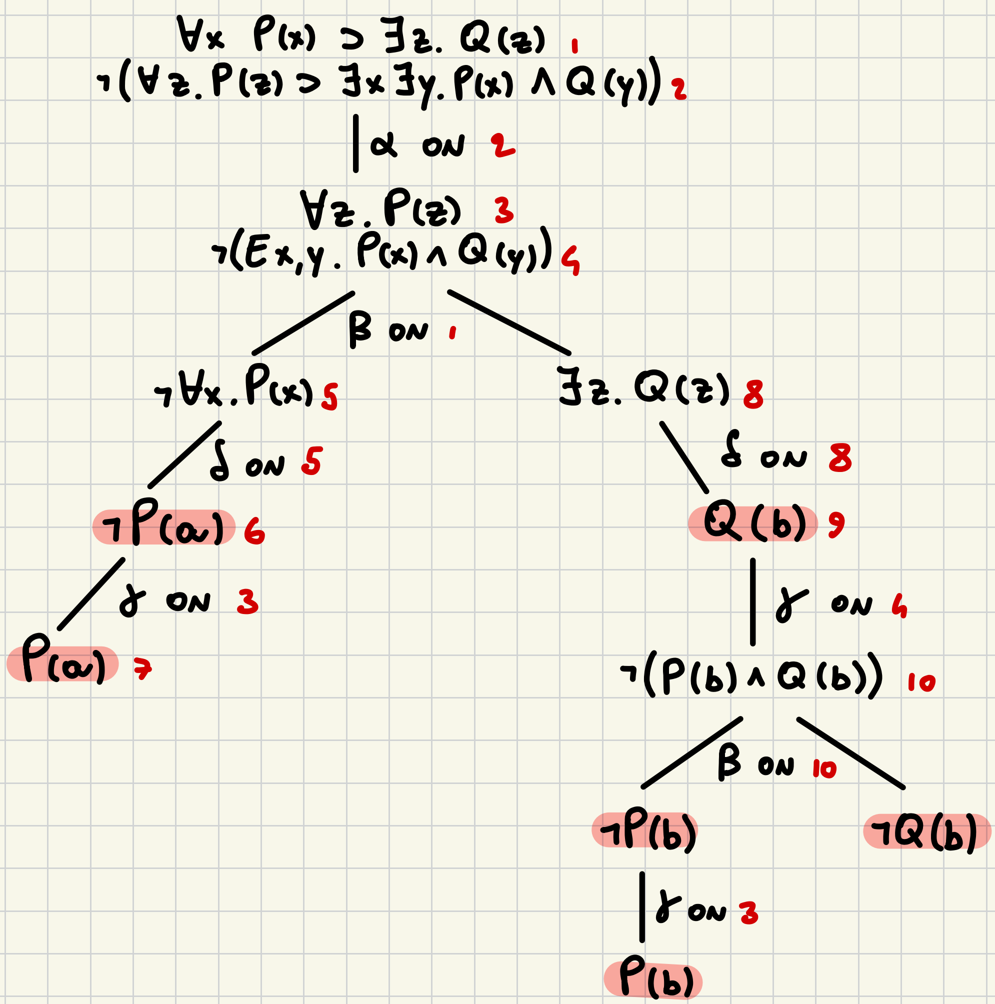
RETURNED PLAN: REFUEL() MOVE (EARTH, JUPITER) MOVE (JUPITER, SATURN)

**Exercise 3.**

1. Using the tableau method, check whether the following holds:

$$\forall x.P(x) \supset \exists z.Q(z) \models \forall z.P(z) \supset \exists x \exists y P(x) \wedge Q(y)$$

   If this is not the case, show a counter-model obtained from the tableau.




THE LOGICAL IMPLICATION HOLDS