SAPIENZA Università di Roma – MSc. in Engineering in Computer Science
Artificial Intelligence & Machine Learning – April 5, 2024

Part 1 - Artificial Intelligence
(Time to complete the test: 2:30 hours)

Suppose we have solar-powered cars that can move to specific locations in the city to deliver orders.

Fluents:

- $AtLoc(x, y)$ denotes that car x is in the location y.
- $Delivered(y)$ denotes that the order has been delivered in location y.
- $Empty(x)$ denotes that the battery level of car x is empty.

Actions:

- $move(x, y)$, which allows car x to move in location y. The action can be done if the battery of x is not empty and if no order has been delivered in y. The effect is that x is in location y and not anymore in the previous one.
- $recharge(x)$, which allows car x to recharge its battery using the solar energy. The action can be done if the battery of x is empty, and the effect is that the battery of x will be full again.
- $deliver(x, y)$, which allows car x to deliver an order in location y. The action can be done if x is located in y, and the effect is that the order is delivered in y and the battery of x becomes empty.

Initial situation:
We have three cars $c1$, $c2$ and $c3$ with a full battery, and four locations $l1$, $l2$, $l3$ and $l4$.
Initially, the cars are in $l2$, and no order has been delivered in any location.

Exercise 1. Formalize the above scenario as a Basic Action Theory in Situation Calculus. Consider the sequence of actions $\varrho = move(c1, l3); deliver(c1, l3); recharge(c1)$:

1. Check by regression if ϱ is executable in S_0;
2. Check by regression whether ϱ results in a situation where $c1$ is in $l2$.
3. Check by regression whether ϱ results in a situation where the order has been delivered in $l3$.

Exercise 2. Consider the following goal $AtLoc(c1, l3) \wedge \neg AtLoc(c1, l1) \wedge \neg AtLoc(c1, l2) \wedge \neg AtLoc(c1, l4)$. First formalize the above scenario as a PDDL domain file and a PDDL problem file. Then:

1. Draw the corresponding transition system;
2. Solve planning for achieving the above goal by using forward depth-first search (uninformed), reporting the various steps of the forward search computation, and returning the resulting plan.

Exercise 3.

1. Formalize the following knowledge in FOL, by introducing suitable predicates, functions and constants, as needed:
   - (ϕ_1) Every animal is carnivore or herbivore
   - (ϕ_2) Carnivores are not herbivores
   - (ϕ_3) Only carnivores chase other animals
   - (ϕ_4) No animal can chase itself

   Using the tableaux method, check whether the knowledge base $KB = \{\phi_1, \phi_2, \phi_3\}$ logically implies ϕ_4. If not, show a counterexample.

2. Consider following propositional knowledge base KB:
   - $\neg a \vee b \vee c$
   - $\neg b \vee a$
   - $\neg c \vee a$
   - $\neg b \vee \neg c$
   - $\neg a \vee c$

   Using the DPLL procedure, check whether $KB \models c \supset a$.

Exercise 1. Formalize the above scenario as a Basic Action Theory in Situation Calculus. Consider the sequence of actions $\varrho = move(c1, l3); deliver(c1, l3); recharge(c1)$:

1. Check by regression if ϱ is executable in S_0;
2. Check by regression whether ϱ results in a situation where $c1$ is in $l2$.
3. Check by regression whether ϱ results in a situation where the order has been delivered in $l3$.

BASIC ACTION THEORY

PRECONDITION AXIOMS

$$Poss(MOVE(x,y), s) \equiv \neg EMPTY(x,s) \wedge \neg DELIVERED(y,s)$$
$$Poss(RECHARGE(x), s) \equiv EMPTY(x,s)$$
$$Poss(DELIVER(x,y), s) \equiv ATLOC(x,y,s)$$

SUCCESSOR STATE AXIOMS

EFFECT AXIOMS

$$a = MOVE(x,y) \supset \exists z. (ATLOC(x,z,s) \supset \neg ATLOC(x,z,DO(a,s))) \wedge ATLOC(x,y,DO(a,s))$$

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

$$a = DELIVER(x,y) \supset DELIVERED(y, DO(a,s)) \wedge EMPTY(x, DO(a,s))$$

NORMALIZE

$$\exists y. (a = MOVE(x,y) \wedge ATLOC(x,z,s)) \supset \neg ATLOC(x,z,DO(a,s))$$

$$a = MOVE(x,y) \supset ATLOC(x,y,DO(a,s))$$

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

$$\exists x. a = DELIVER(x,y) \supset DELIVERED(y, DO(a,s))$$

$$\exists y. a = DELIVER(x,y) \supset EMPTY(x, DO(a,s))$$

EXPLANATION CLOSURE

$$ATLOC(x,y,DO(a,s)) \equiv (a = MOVE(x,y)) \vee (ATLOC(x,y,s) \wedge \neg \exists z. a = MOVE(x,z))$$

$$EMPTY(x, DO(a,s)) \equiv (\exists y. a = DELIVER(x,y)) \vee (EMPTY(x,s) \wedge \neg a = RECHARGE(x))$$

$$DELIVERED(y, DO(a,s)) \equiv \exists x. a = DELIVER(x,y) \vee DELIVERED(y,s)$$

INITIAL SITUATION

$\neg EMPTY(c_1, S_0)$ $\neg EMPTY(c_2, S_0)$ $\neg EMPTY(c_3, S_0)$

$ATLOC(c_1, l_2, S_0)$ $ATLOC(c_2, l_2, S_0)$ $ATLOC(c_3, l_2, S_0)$

$\neg DELIVERED(l_1, S_0)$ $\neg DELIVERED(l_2, S_0)$ $\neg DELIVERED(l_3, S_0)$ $\neg DELIVERED(l_4, S_0)$

1) $\varrho = move(c1, l3); deliver(c1, l3); recharge(c1)$ EXECUTABLE IN S_0?

$S_1 = DO(MOVE(c_1, l_3), S_0)$ 1. $D_0 \cup D_{una} \models R[Poss(MOVE(c_1, l_3), S_0)]$

$S_2 = DO(DELIVER(c_1, l_3), S_1)$ 2. $D_0 \cup D_{una} \models R[Poss(DELIVER(c_1, l_3), S_1)]$

$S_3 = DO(RECHARGE(c_1), S_2)$ 3. $D_0 \cup D_{una} \models R[Poss(RECHARGE(c_1), S_2)]$

1. $R[Poss(MOVE(c_1, l_3), S_0)] = R[\neg EMPTY(c_1, S_0) \wedge \neg DELIVERED(l_3, S_0)] =$
$= \neg EMPTY(c_1, S_0) \wedge \neg DELIVERED(l_3, S_0) =$ TRUE

MOVE (c_1, l_3) IS EXECUTABLE IN S_0 ✓

2. $R[Poss(DELIVER(c_1, l_3), S_1)] = R[ATLOC(c_1, l_3, S_1)] =$
$= R[ATLOC(c_1, l_3, DO(MOVE(c_1, l_3), S_0))] =$
$= R[(MOVE(c_1, l_3) = MOVE(c_1, l_3)) \vee (ATLOC(c_1, l_3, S_0) \wedge$
$\exists z.\, MOVE(c_1, l_3) \neq MOVE(c_1, z))] =$
$= R[TRUE] =$ TRUE

DELIVER (c_1, l_3) IS EXECUTABLE IN S_1 ✓

3. $R[Poss(RECHARGE(c_1), S_2)] = R[EMPTY(c_1, S_2)] =$
$= R[EMPTY(c_1, DO(DELIVER(c_1, l_3), S_1))] =$
$= R[\exists y.(DELIVER(c_1, l_3) = DELIVER(c_1, y)) \vee (EMPTY(c_1, S_1) \wedge$
$DELIVER(c_1, l_3) = RECHARGE(c_1))] =$
$= R[TRUE] =$ TRUE

RECHARGE $(c_1), S_2$ IS EXECUTABLE IN S_2 ✓

ρ IS EXECUTABLE IN S_0 ✓

2) c_1 IS IN l_2 $\rightarrow$ $D_0 \cup D_{una} \models \mathcal{R}[AtLoc(c_1, l_2, S_3)]$=

$\mathcal{R}[AtLoc(c_1, l_2, S_3)] = \mathcal{R}[AtLoc(c_1, l_2, do(RECHARGE(c_1), S_2))]$=
$= \mathcal{R}[(RECHARGE(c_1) = move(c_1, l_2)) \vee (AtLoc(c_1, l_2, S_2)$
$\quad \wedge \exists z. RECHARGE(c_1) \neq move(c_1, z))]$=
$= \mathcal{R}[AtLoc(c_1, l_2, S_2)]$=
$= \mathcal{R}[AtLoc(c_1, l_2, do(DELIVER(c_1, l_3), S_1))]$=
$= \mathcal{R}[(DELIVER(c_1, l_3) = move(c_1, l_2)) \vee (AtLoc(c_1, l_2, S_1)$
$\quad \wedge \exists z. DELIVER(c_1, l_3) \neq move(c_1, z))]$=
$= \mathcal{R}[AtLoc(c_1, l_2, S_1)]$=
$= \mathcal{R}[AtLoc(c_1, l_2, do(move(c_1, l_3), S_0))]$=
$= \mathcal{R}[(move(c_1, l_3) = move(c_1, l_2)) \vee (AtLoc(c_1, l_2, S_0) \wedge$
$\quad \exists z. move(c_1, l_3) \neq move(c_1, z))]$=
$= \mathcal{R}[FALSE]$ = FALSE

3) $D_0 \cup D_{una} \models \mathcal{R}[DELIVERED(l_3, S_3)]$

$\mathcal{R}[DELIVERED(l_3, S_3)] = \mathcal{R}[DELIVERED(l_3, do(RECHARGE(c_1), S_2))]$=
$= \mathcal{R}[(\exists x. RECHARGE(c_1) = DELIVER(x, l_3)) \vee DELIVERED(l_3, S_2)]$=
$= \mathcal{R}[DELIVERED(l_3, S_2)]$=
$= \mathcal{R}[DELIVERED(l_3, do(DELIVER(c_1, l_3), S_1))]$=
$= \mathcal{R}[(\exists x. DELIVER(c_1, l_3) = DELIVER(x, l_3)) \vee DELIVERED(l_3, S_1)]$=
$= \mathcal{R}[TRUE]$ = TRUE

Exercise 2. Consider the following goal $AtLoc(c1, l3) \wedge \neg AtLoc(c1, l1) \wedge \neg AtLoc(c1, l2) \wedge \neg AtLoc(c1, l4)$. First formalize the above scenario as a PDDL domain file and a PDDL problem file. Then:

1. Draw the corresponding transition system;
2. Solve planning for achieving the above goal by using forward depth-first search (uninformed), reporting the various steps of the forward search computation, and returning the resulting plan.

```
(DEFINE (DOMAIN CAR.DOM)
  (:REQUIREMENTS :ADL)
  (:TYPES CAR LOC)
  (:PREDICATES
      (AtLoc ?x -CAR ?y-LOC)
      (DELIVERED ?y -LOC)
      (EMPTY ?x -CAR)
  )
  (:ACTION MOVE
     :PARAMETERS (?x -CAR ?y-LOC)
     :PRECONDITIONS(AND (NOT (EMPTY ?x)) (NOT (DELIVERED ?y)))
     :EFFECT(AND (EXISTS (?z-LOC) (WHEN (ATLOC ?x ?z)
                                     (NOT (AtLoc ?x ?z))))
                 (AtLoc ?x ?y))
  ); END OF MOVE
  (:ACTION RECHARGE
     :PARAMETERS ( ?x -CAR)
     :PRECONDITIONS (EMPTY ?x)
     :EFFECT (NOT (EMPTY ?x))
  ); END OF RECHARGE
  (:ACTION DELIVER
     PARAMETERS ( ?x-CAR ?y-LOC)
     :PRECONDITIONS (AT LOC ?x ?y)
     :EFFECT (AND(DELIVERED ?y) (EMPTY ?x))
  ); END OF DELIVER
); END OF DEF DOM

(DEFINE (PROBLEM CAR.PROB)(:DOMAIN CAR.DOM)
  (:OBJECTS    l1 l2 l3 l4 -LOC    c1 c2 c3 -CAR
  )
  (:INIT  (AND ( FORALL (?y -LOC)
                    (NOT (DELIVERED ?y))
                )
                (FORALL (?x -CAR)
                    (AND
                      (NOT (EMPTY ?x))
                      (AtLoc ?x l2))
                )
          )
  )
  (:GOAL (AND(AtLoc c1 l3)(NOT (AtLoc c1 l1))(NOT (AtLoc c1 l2))(NOT (AtLoc c1 l4)))
); END OF DEF
```

2) 0. $\mathcal{T} = [(S_0, \text{EMPTY})]$
$m = \{S_0\}$

1. $(\text{STATE}, \text{PLAN}) = \mathcal{T}.\text{POP}()$
$m.\text{ADD}(S_1)$
$\mathcal{T}.\text{PUSH}(S_1, \text{MOVE}(c_1, l_3))$

$\mathcal{T} = [(S_1, \text{MOVE}(c_1, l_3))]$ $\quad m = \{S_0, S_1\}$

2. $(\text{STATE}, \text{PLAN}) = \mathcal{T}.\text{POP}()$

RETURNE PLAN: $\text{MOVE}(c_1, l_3)$

Exercise 3.

1. Formalize the following knowledge in FOL, by introducing suitable predicates, functions and constants, as needed:
   - (ϕ_1) Every animal is carnivore or herbivore
   - (ϕ_2) Carnivores are not herbivores
   - (ϕ_3) Only carnivores chase other animals
   - (ϕ_4) No animal can chase itself

   Using the tableaux method, check whether the knowledge base $KB = \{\phi_1, \phi_2, \phi_3\}$ logically implies ϕ_4. If not, show a counterexample.

2. Consider following propositional knowledge base KB:
   - $\neg a \vee b \vee c$
   - $\neg b \vee a$
   - $\neg c \vee a$
   - $\neg b \vee \neg c$
   - $\neg a \vee c$

   Using the DPLL procedure, check whether $KB \models c \supset a$.

2) $KB \models \neg(c \supset a) \rightarrow KB \models c \wedge \neg a$

$\phi: \{\{\neg a, b, c\}, \{\neg b, a\}, \{\neg c, a\}, \{\neg b, \neg c\}, \{\neg a, c\}, \{c\}, \{\neg a\}\}$

UP) $a = F \rightarrow \phi: \{\{\neg b\}, \{\neg c\}, \{\neg b, \neg c\}, \{c\}\}$

UP) $b = F \rightarrow \phi: \{\{\neg c\}, \{c\}\}$

UP) $c = F \rightarrow \phi: \{\{\}\}$ NOT SATISFIABLE

1) $\phi_1: \forall x. (\text{ANIMAL}(x) \supset (\text{CARNIVORE}(x) \vee \text{HERBIVORE}(x)))$

$\phi_2: \forall x. (\text{CARNIVORE}(x) \supset \neg \text{HERBIVORE}(x))$

$\phi_3: \forall x, y. (\text{CHASE}(x, y) \supset \text{CARNIVORE}(x))$

$\phi_4: \forall x. (\text{ANIMAL}(x) \supset \neg \text{CHASE}(x, x))$

$\{\phi_1, \phi_2, \phi_3\} \models \phi_4$

$\forall x. (\text{ANIMAL}(x) \supset (\text{CARNIVORE}(x) \vee \text{HERBIVORE}(x)))$ 1

$\forall x. (\text{CARNIVORE}(x) \supset \neg \text{HERBIVORE}(x))$ 2

$\forall x, y. (\text{CHASE}(x, y) \supset \text{CARNIVORE}(x))$ 3

$\neg \forall x. (\text{ANIMAL}(x) \supset \neg \text{CHASE}(x, x))$ 4

δ ON 4

$\neg(\text{ANIMAL}(a) \supset \neg \text{CHASE}(a, a))$ 5

α ON 5

$\text{ANIMAL}(a)$ 6

$\text{CHASE}(a, a)$ 7

γ ON 1

$\neg \text{ANIMAL}(a) \vee (\text{CARNIVORE}(a) \vee \text{HERBIVORE}(a))$ 8

β ON 8

$\neg \text{ANIMAL}(a)$ 9 — $\times$

$(\text{CARNIVORE}(a) \vee \text{HERBIVORE}(a))$ 10

β ON 10

$\text{CARNIVORE}(a)$ 11

$\text{HERBIVORE}(a)$ 12

γ ON 3

$\neg \text{CHASE}(a, a) \vee \text{CARNIVORE}(a)$ 16

γ ON 2

$\neg \text{CARNIVORE}(a) \vee \neg \text{HERBIVORE}(a)$ 13

β ON 16

$\neg \text{CHASE}(a, a)$ 17 — $\times$

$\text{CARNIVORE}(a)$ 18

β ON 13

$\neg \text{CARNIVORE}(a)$ 14

$\neg \text{HERBIVORE}(a)$ 15 — $\times$

γ ON 3

$\neg \text{CHASE}(a, a) \vee \text{CARNIVORE}(a)$ 19

β ON 19

$\neg \text{CHASE}(a, a)$ 20 — $\times$

$\text{CARNIVORE}(a)$ 21 — $\times$

NOT SATISFIABLE, SO $\rightarrow \{\phi_1, \phi_2, \phi_3\} \not\models \phi_4$