SAPIENZA Università di Roma – MSc. in Engineering in Computer Science
Artificial Intelligence & Machine Learning – Feb 24, 2025

Part 1 - Artificial Intelligence
(Time to complete the test: 2:00 hours)

Consider a simplified Pac-Man world consisting in a 3x3 grid, as shown below:

	A	*B*	*C*
1			
2		■	
3	●	■	

Besides Pac-Man, the grid contains: power pellets (●); obstacles (■); ghosts.

Consider the following formalization of the scenario:

Non-Fluents:

- *Connected*(c_1, c_2), denoting that cells c_1 and c_2 are connected: two cells are connected if they share an *entire* side (not just a corner); for instance, $B1$ is connected to $C1$ but not to $C2$; notice also that *Connected* is symmetrical, i.e., c_1 is connected to c_2 if and only if c_2 is connected to c_1; cells are not connected to themselves.
- *ObstacleIn*(c), denoting that cell c contains an obstacle.

Fluents:

- *In*(c), denoting that Pac-Man is in cell c.
- *Pellet*(c), denoting that cell c contains a power pellet.
- *Super*$()$, denoting that Pac-Man has superpowers.
- *Alive*$()$, denoting that Pac Man is alive.
- *GhostIn*(c), denoting that cell c contains a ghost (notice that ghosts are static).

Actions:

- *move*(c), which allows Pac-Man to move to cell c. The action can be done only if Pac-Man is alive, in a cell c' connected to c, and c contains no obstacles (it may contain a ghost, though). The effect is that: 1. if there are no ghosts in c or Pac-Man has superpowers then Pac-Man is in cell c; 2. any power pellet in c disappears; 3. if there is a power pellet in c then Pac-Man has superpowers (once obtained, superpowers last forever); 4. if there is a ghost in c and Pac-Man does not have superpowers, then Pac-Man is no longer alive and disappears from the grid.
- *eatGhost*$()$, which allows Pac-Man to eat a ghost. The action can be done only if Pac-Man has superpowers and is in a cell containing a ghost. The effect is that the ghost disappears from the cell.

Initial situation:
Pac-Man is alive. Cells, connections, Pac-Man's initial position, power pellets, and ghosts are as in the figure above.

Exercise 1.

1. Formalize the above scenario as a Basic Action Theory.
2. Using regression, check whether the action sequence $\varrho = move(A2); move(A3);$ leads to a situation where Pac-Man has superpowers.

Exercise 2.

Considering goal $\gamma = In(C3)$,

1. Formalize the above scenario as a PDDL domain file and a PDDL problem file.
2. Draw the corresponding entire transition system;
3. Solve planning for achieving γ, using forward depth-first search (uninformed), reporting the steps of the forward search computation, and returning the resulting plan. **The search must be exhaustive**, i.e., all states must be visited.

Exercise 3.

1. Formalize the following sentences in First-Order Logic:
    - ϕ_1: Every beverage is liquid
    - ϕ_2: Not all liquids are beverages
    - ϕ_3: Some liquids are dangerous
    - ϕ_5: There exists exactly one dangerous beverage
2. Given the following formulas:
    - $\phi_1 = (\forall x.P(x) \supset Z(x)) \wedge (\forall x.Q(x) \supset Z(x))$
    - $\phi_2 = \forall x.(Z(x) \wedge P(x)) \supset \neg Q(x)$
    - $\phi_3 = \neg\exists x.P(x) \wedge Q(x)$

    use the Tableaux method to check whether $\{\phi_1, \phi_2\} \models \phi_3$

Exercise 1.

1. Formalize the above scenario as a Basic Action Theory.
2. Using regression, check whether the action sequence $\varrho = move(A2); move(A3);$ leads to a situation where Pac-Man has superpowers.

1) PRECONDITIONS AXIOMS

$$Poss(MOVE(c), s) \equiv \exists c'. IN(c', s) \wedge CONNECTED(c', c) \wedge ALIVE(s) \wedge \neg OBSTACLEIN(c, s)$$

$$Poss(EATGHOST(), s) \equiv \exists c. IN(c, S) \wedge GHOSTIN(c, S) \wedge SUPER(s)$$

SUCCESSOR STATE AXIOMS

EFFECT AXIOMS

$$a = MOVE(c) \supset (\neg GHOSTIN(c, s) \supset IN(c, DO(a, s))) \wedge (SUPER(s) \supset IN(c, DO(a, s))) \wedge \neg PELLET(c, DO(a, s)) \wedge (PELLET(c, s) \supset SUPER(DO(a, s))) \wedge (GHOSTIN(c, S) \wedge \neg SUPER(s) \supset \neg ALIVE(DO(a, s)) \wedge (\forall x. IN(x, S) \supset \neg IN(x, DO(a, s))))$$

$$a = EATGHOST() \supset Ec.(GHOSTIN(c, s) \supset \neg GHOSTIN(c, DO(a, s)))$$

NORMALIZE

$$(a = MOVE(c) \wedge \neg GHOSTIN(c, S)) \supset IN(c, DO(a, s))$$

$$(a = MOVE(c) \wedge SUPER(s)) \supset IN(c, DO(a, s))$$

$$a = MOVE(c) \supset \neg PELLET(c, DO(a, s))$$

$$\exists c.(a = MOVE(c) \wedge PELLET(c, s)) \supset SUPER(DO(a, s)))$$

$$\exists c.(a = MOVE(c) \wedge GHOSTIN(c, S) \wedge \neg SUPER(s)) \supset \neg ALIVE(DO(a, S))$$

$$\exists c.(a = MOVE(c) \wedge GHOSTIN(c, S) \wedge \neg SUPER(s) \wedge IN(x, s)) \supset \neg IN(x, DO(a, s))$$

$$(a = EATGHOST() \wedge GHOSTIN(c, s)) \supset \neg GHOSTIN(c, DO(a, s))$$

EXPLANATION CLOSURE

$$IN(c, DO(a, s)) \equiv (a = MOVE(c) \wedge \neg GHOSTIN(c, S)) \vee (a = MOVE(c) \wedge SUPER(s)) \vee (IN(c, S) \wedge \neg(\exists c'.(a = MOVE(c') \wedge GHOSTIN(c', S) \wedge \neg SUPER(s))))$$

$$PELLET(c, DO(a, s)) \equiv (PELLET(c, s) \wedge \neg\, a = MOVE(c))$$

$$SUPER(DO(a, s)) \equiv \exists c.(a = MOVE(c) \wedge PELLET(c, s)) \vee SUPER(s)$$

$ALIVE(DO(a,s)) \equiv (ALIVE(s) \wedge \neg(\exists c.\ a = MOVE(c) \wedge GHOSTIN(c,s) \wedge \neg SUPER(s)))$

$GHOSTIN(c, DO(a,s)) \equiv (GHOSTIN(c,s) \wedge \neg a = EATGHOST(c))$

INITIAL SITUATION

$ALIVE(S_0)$
$CONNECTED(c_1, c_2) \equiv (c_1 = A_1,\ c_2 = B_1) \ldots\ldots$
$IN(c, S_0) \equiv (c = A_1)$
$PELLET(c, S_0) \equiv (c = A_3)$
$GHOSTIN(c, S_0) \equiv (c = C_2)$
$OBSTACLES(c) \equiv (c = B_2)\ (c = B_3)$

2) $\varrho = move(A2); move(A3);$

$S_1 = DO(MOVE(A_2), S_0)$
$S_2 = DO(MOVE(A_3), S_1)$

$D_0 \cup D_{UNA} \models R[SUPER(S_2)]$

$R[SUPER(S_2)] = R[SUPER(DO(MOVE(A_3), S_1))] =$
$= R[\exists c.(MOVE(A_3) = MOVE(c) \wedge PELLET(c, S_1)) \vee SUPER(S_1)] =$
$= R[PELLET(S_1)] \vee R[SUPER(S_1)] =$

$R[PELLET(S_1)] = R[PELLET(DO(MOVE(A_2), S_0))] =$
$= R[PELLET(c, S_0) \wedge MOVE(A_2) \neq MOVE(c)] =$
$= PELLET(c, S_0) \wedge MOVE(A_2) \neq MOVE(c)$

$R[SUPER(S_1)] = R[SUPER(DO(MOVE(A_2), S_0))] =$
$= R[\exists c.(MOVE(A_2) = MOVE(c) \wedge PELLET(c, S_0)) \vee SUPER(S_0)] =$
$= R[FALSE] =$ FALSE

$R[SUPER(S_2)] = R[PELLET(S_1)] \vee R[SUPER(S_1)]$

TRUE FOR $c = A_3 \rightarrow$ ρ LEADS TO A SITUATION WHERE ᗧ IS SUPER ✓

Exercise 2.

Considering goal $\gamma = In(C3)$,

1. Formalize the above scenario as a PDDL domain file and a PDDL problem file.
2. Draw the corresponding entire transition system;
3. Solve planning for achieving γ, using forward depth-first search (uninformed), reporting the steps of the forward search computation, and returning the resulting plan. **The search must be exhaustive**, i.e., all states must be visited.

1)
```
(DEFINE (DOMAIN PAC_DOM)
  (:REQUIREMENTS :ADL)
  (:TYPES CELL)
  (: PREDICATES
      (CONNECTED ?C1 ?C2 -CELL)
      (OBSTACLEIN ?C -CELL)
      (IN ?C - CELL)
      (PELLET ?C - CELL)
      (SUPER)
      (ALIVE)
      (GHOSTIN ?C - CELL)
  )
  (: ACTION MOVE
    :PARAMETERS ( ?C - CELL)
    : PRECONDITIONS (AND
                      (EXISTS (?C' - CELL) (IN ?C') (CONNECTED ?C' ?C)
                       (ALIVE) (NOT (OBSTACLEIN ?C)))
                      )
    :EFFECT (AND
              (WHEN (NOT (GHOSTIN ?C)) (IN ?C))
              (WHEN (SUPER) (IN ?C))
              (NOT (PELLET ?C))
              (WHEN (PELLET ?C) ( SUPER))
              (WHEN (AND (GHOSTIN ?C) (NOT (SUPER)))
                    (AND (NOT (ALIVE))
                         (FORALL (?X -CELL)
                            (WHEN (IN ?X) (NOT ( IN ?X))
                            )
                         )
                    )
              )
            )
  ); END OF MOVE
  (: ACTION EAT GHOST
    : PARAMETERS ()
    : PRECONDITION (EXIST (?C - CELL) (AND ( IN ?C) (GHOSTIN ?C) (SUPER)))
    : EFFECT (EXISTS ( ?C - CELL) (WHEN (GHOSTIN ?C) (NOT (GHOSTIN ?C))))
  ); END OF EAT GHOST
); END OF DEFINE DOM
```

```
(DEFINE (PROBLEM PAC.PROB)(:DOMAIN PAC.DOM)
  (:OBJECTS  A1 A2 A3  B1 B2 B3  C1 C2 C3 - CELL)
  (:INIT  (ALIVE) (CONNECTED A1 B1)....... (IN A1)
          (OBSTACLEIN B2)(OBSTACLEIN B3) (PELLET A3) (GHOSTIN A1)
  )
  (:GOAL  (IN C3)
  )
); END OF DEF PROB
```

Exercise 3.

1. Formalize the following sentences in First-Order Logic:
   - ϕ_1: Every beverage is liquid
   - ϕ_2: Not all liquids are beverages
   - ϕ_3: Some liquids are dangerous
   - ϕ_5: There exists exactly one dangerous beverage
2. Given the following formulas:
   - $\phi_1 = (\forall x.P(x) \supset Z(x)) \wedge (\forall x.Q(x) \supset Z(x))$
   - $\phi_2 = \forall x.(Z(x) \wedge P(x)) \supset \neg Q(x)$
   - $\phi_3 = \neg \exists x.P(x) \wedge Q(x)$

   use the Tableaux method to check whether $\{\phi_1, \phi_2\} \models \phi_3$

1) ϕ_1: $\forall x.$ BEVERAGE$(x) \supset$ LIQUID(x)

ϕ_2. $\exists x.$ LIQUID$(x) \wedge \neg$ BEVERAGE(x)

ϕ_3: $\exists x.$ LIQUID$(x) \wedge$ DANGEROUS(x)

ϕ_4: $\exists x.($BEVERAGE$(x) \wedge$ DANGEROUS$(x)) \wedge \forall y.\,($BEVERAGE$(y) \wedge$ DANGEROUS$(y) \supset y=x)$

2)

$(\forall x. P(x) \supset Z(x)) \land (\forall x. Q(x) \supset Z(x))$ 1

$\forall x. (Z(x) \land P(x)) \supset \neg Q(x)$ 2

$\exists x. P(x) \land Q(x)$ 3

δ ON 3

$P(a) \land Q(a)$ 4

α ON 4

$P(a)$ 5

$Q(a)$ 6

γ ON 2

$\neg(Z(a) \land P(a)) \lor \neg Q(a)$ 7

β ON 7

$\neg Z(a) \lor \neg P(a)$ 8

$\neg Q(a)$ 9

$\times$

β ON 8

$\neg Z(a)$ 9

$\neg P(a)$ 10

$\times$

γ ON 1

$(P(a) \supset Z(a)) \land (Q(a) \supset Z(a))$ 11

α ON 11

$\neg P(a) \lor Z(a)$ 12

$\neg Q(a) \lor Z(a)$ 13

β ON 12

$\neg P(a)$

$\times$

$Z(a)$

$(\phi_1, \phi_2) \nvDash \neg \phi_3 \longrightarrow \{\phi_1, \phi_2\} \vDash \phi_3$