**PDDL**

1. **Please state in your own words what the Star Puzzle problem is (5 points).**

Ans:

The Star Puzzle provides an algorithm to solve the Tower of Hanoi problem using 4 posts. In the Star Puzzle, the position of posts are as followed: 3 posts labelled A,B,C form an equilateral triangle and a fourth post labelled O is placed at the centre. The restriction imposed due to this arrangement is that every disk has to move between posts A,B,C via O, direct transfer is not allowed. Hence, the allowed moves are in the form of a Star.

**Goal:** To transfer n disks from post A to C keeping the size constraints that no bigger disk can be placed on top of a smaller disk restricting the allowable moves as per above discussion.

**Algorithm:**

Iterating over i: [1,n]

1. Recursively transfer (n-i) smaller disks from post A to B using all 4 posts
2. Transfer the largest disk from A to C using three-in-a-row algorithm (Three-in-a-row algorithm states that allowable transfer is from A to O, then O to B, direct A to B is not possible)
3. Recursively transfer the smallest n-i disks from B to B using all 4 posts.

**Analysis:**

If S(n) is the minimum moves made using this algorithm, then for n>=1 we get:

S(n)=min(2S(n-i) + 3i – 1)

Also it is the presumed optimal algorithm for the Tower of Hanoi puzzle.

1. **Looking at the code above, explain briefly what is the usage of each predicate, i.e. what is their purpose (10 points).**

Ans: The predicates explain the conditions we might need to use while validating our actions.

The usage /purpose of the predicates are:

1. (post\_o ?o) : This predicate checks if the variable o is at Post O or not. Because post O is the allowable to and from post for each of the other 3 posts A,B,C.
2. (on ?x ?y) : This predicate checks if a variable x is present on top of y or not. Here variable x,y refer to disks or a disk and a post.
3. (smaller ?s ?l) : This predicate checks if the variable (disk) s is less than variable(disk) l or not.
4. (clear ?z) : This predicate checks if the current state of a post is clear to move or to be moved from a post or not.
5. **Explain the functionality of each action and the way they operate. How does this relate to the PDDL organization of declarative and procedural knowledge (15 points).**

Ans:

1. Action **MOVETO**

MOVETO describes the functionality of moving one disk from one post to post O. The parameters denote **Declarative knowledge** and the parameters involved are:

1. The currently processed disk
2. The post from where to move
3. The post to which to move to.

The precondition denotes the **Procedural Knowledge** since we are describing how we define a precondition. The Precondition states a conjunction of the following conditions:

1. Is the destination post the same as Post O?
2. Is the currently processed disk smaller than the post to which it is transferred or smaller than the disk present there
3. Is the currently processed disk on top of the source post mentioned as parameter.
4. Is the disk clear to be moved?
5. Is the destination post clear to be moved to?

The effect checks the conditions once the transfer action is completed and it denotes the **Procedural Knowledge** since we are describing how we define an effect. It states a conjunction of the following conditions:

1. The currently processed disk is on top of the destination post(post O)
2. If the source post is cleared
3. The disk is no longer present on the source post
4. Asserts that Destination Post is no longer Post O( since action is already completed)
5. Assigns Post O as the source post to continue next transfers.
6. Assigns that Destination Post is no longer clear
7. Action **MOVEFROM**

MOVEFROM describes the functionality of moving one disk from post O to one of the posts A,B,C.

The parameters denote **Declarative knowledge** and the parameters involved are:

1. The currently processed disk
2. The post from where to move
3. The post to which to move to.

The precondition denotes the **Procedural Knowledge** since we are describing how we define a precondition. The Precondition states a conjunction of the following conditions:

1. Is the disk at Post O?
2. Is the currently processed disk smaller than the post to which it is transferred or smaller than the disk present there
3. Is the currently processed disk on top of the source post mentioned as parameter.
4. Is the disk clear to be moved?
5. Is the destination post clear to be moved to?

The effect checks the conditions once the transfer action is completed and it denotes the **Procedural Knowledge** since we are describing how we define an effect. It states a conjunction of the following conditions:

1. The currently processed disk is on top of the destination post(post A/B/C)
2. If the source post is cleared
3. The disk is no longer present on the source post
4. Asserts that disk is no longer on Post O( since action is already completed)
5. Assigns Post O as the source post to continue next transfers.
6. Assigns that Destination Post is no longer clear
7. **This is a concrete instantiation of the domain where we have 4 disks. Using this concrete instantiation, explain the objects, locations and the goal (15 points).**
8. **Objects:**

Objects consists of the 4 posts A,B,C,O as PA, PB, PC, PO and the 4 disks since its n=4, namely D1, D2, D3, D4.

1. **Locations:**

Location puts constraints on the allowable locations.

It says the following:

1. All the disks D1, D2, D3, D4 are smaller than all the posts PA, PB, PC and PO.
2. Disk D1 is smaller than all other disks D2, D3, D4.
3. Disk D2 is smaller than D3, D4
4. Disk D3 is smaller than D4.
5. It also assigns a clear tag to posts PB, PC and PO denoting that these posts are cleared to be moved to
6. It assigns a clear tag to disk D1 denoting that this disk is clear to move to a new position.
7. Finally it assigns the initial position saying D4 is on post PA, D3 is on D4, D2 is on D3 and D1 is on D2.
8. **Goal:**

The final goal stated is that:

1. D4 is on post PC, D3 is on D4, D2 is on D3 and D1 is on D2.(I.e. all disks are transferred to Post C and arranged in the increasing of their size from top to bottom)

5. **What is the resulting plan for solving the problem with 4 disks? (5 points)**

Ans:

(moveto d1 d2 po)

(movefrom d1 po pb)

(moveto d2 d3 po)

(movefrom d2 po pc)

(moveto d3 d4 po)

(moveto d2 pc d3)

(moveto d1 pb d2)

(movefrom d1 d2 d4)

(movefrom d2 d3 pb)

(movefrom d3 po pc)

(moveto d1 d4 po)

(movefrom d1 po d2)

(moveto d4 pa po)

(moveto d3 pc d4)

(movefrom d3 d4 pa)

(movefrom d4 po pc)

(moveto d3 pa po)

(movefrom d3 po d4)

(moveto d1 d2 po)

(movefrom d1 po d3)

(moveto d2 pb po)

(moveto d1 d3 d2)

(movefrom d1 d2 pb)

(movefrom d2 po d3)

(moveto d1 pb po)

(movefrom d1 po d2)

1. **Now you should be able to instantiate the problem, i.e. making the planning instances (in PDDL) for 5 and 6 disks. If your specification is correct, the planner should output a valid plan to solve the problem. You can get 80% of the implementation credit, if the generated plan solves the problem for N disks in Ω(N²) steps. This can be checked from the Plan file generated by your PDDL code. To get full credit, your PDDL description for N disks should consist of O(N) steps (50 points).**

Ans:

1. 5 DISKS

REPORT:

--- OK.

Match tree built with 380 nodes.

PDDL problem description loaded:

Domain: STAR-PUZZLE

Problem: STAR-PUZZLE-4

#Actions: 380

#Fluents: 47

Landmarks found: 5

Starting search with IW (time budget is 60 secs)...

rel\_plan size: 7

#RP\_fluents 17

Caption

{#goals, #UNnachieved, #Achieved} -> IW(max\_w)

{5/5/0}:IW(1) -> [2][3][4][5][6][7][8][9][10];; NOT I-REACHABLE ;;

Total time: 0.004

Nodes generated during search: 47

Nodes expanded during search: 47

IW search completed

Starting search with BFS(novel,land,h\_add)...

--[4294967295 / 9]--

--[1 / 9]--

--[1 / 8]--

--[1 / 7]--

--[1 / 6]--

--[1 / 4]--

--[1 / 2]--

--[1 / 1]--

--[1 / 0]--

--[0 / 0]--

Total time: 0.04

Nodes generated during search: 547

Nodes expanded during search: 187

Plan found with cost: 34

BFS search completed

(moveto d1 d2 po)

(movefrom d1 po pc)

(moveto d2 d3 po)

(movefrom d2 po pb)

(moveto d3 d4 po)

(moveto d2 pb d3)

(movefrom d2 d3 d4)

(movefrom d3 po pb)

(moveto d2 d4 po)

(movefrom d2 po d3)

(moveto d1 pc po)

(movefrom d1 po d2)

(moveto d4 d5 po)

(movefrom d4 po pc)

(moveto d5 pa po)

(moveto d4 pc d5)

(movefrom d4 d5 pa)

(movefrom d5 po pc)

(moveto d4 pa po)

(movefrom d4 po d5)

(moveto d1 d2 po)

(movefrom d1 po d4)

(moveto d2 d3 po)

(movefrom d2 po pa)

(moveto d3 pb po)

(moveto d2 pa d3)

(moveto d1 d4 d2)

(movefrom d1 d2 pb)

(movefrom d2 d3 pa)

(movefrom d3 po d4)

(moveto d2 pa po)

(movefrom d2 po d3)

(moveto d1 pb po)

(movefrom d1 po d2)

1. 6 DISKS

REPORT:

--- OK.

Match tree built with 542 nodes.

PDDL problem description loaded:

Domain: STAR-PUZZLE

Problem: STAR-PUZZLE-4

#Actions: 542

#Fluents: 58

Landmarks found: 6

Starting search with IW (time budget is 60 secs)...

rel\_plan size: 8

#RP\_fluents 20

Caption

{#goals, #UNnachieved, #Achieved} -> IW(max\_w)

{6/6/0}:IW(1) -> [2][3][4][5][6][7][8][9][10];; NOT I-REACHABLE ;;

Total time: 0.008

Nodes generated during search: 47

Nodes expanded during search: 47

IW search completed

Starting search with BFS(novel,land,h\_add)...

--[4294967295 / 10]--

--[1 / 10]--

--[1 / 9]--

--[1 / 8]--

--[1 / 7]--

--[1 / 6]--

--[1 / 4]--

--[1 / 2]--

--[1 / 1]--

--[1 / 0]--

--[0 / 0]--

Total time: 0.072

Nodes generated during search: 706

Nodes expanded during search: 241

Plan found with cost: 52

BFS search completed

PATH:

(moveto d1 d2 po)

(movefrom d1 po pc)

(moveto d2 d3 po)

(movefrom d2 po pb)

(moveto d3 d4 po)

(moveto d2 pb d3)

(movefrom d2 d3 d4)

(movefrom d3 po pb)

(moveto d2 d4 po)

(movefrom d2 po d3)

(moveto d1 pc po)

(movefrom d1 po d2)

(moveto d4 d5 po)

(movefrom d4 po pc)

(moveto d5 d6 po)

(moveto d4 pc d5)

(movefrom d4 d5 d6)

(movefrom d5 po pc)

(moveto d4 d6 po)

(movefrom d4 po d5)

(moveto d6 pa po)

(moveto d4 d5 d6)

(movefrom d4 d6 pa)

(moveto d5 pc d6)

(moveto d4 pa d5)

(movefrom d4 d5 pc)

(movefrom d5 d6 pa)

(moveto d4 pc d6)

(movefrom d4 d6 d5)

(movefrom d6 po pc)

(moveto d4 d5 po)

(movefrom d4 po d6)

(moveto d5 pa po)

(moveto d4 d6 d5)

(movefrom d4 d5 pa)

(movefrom d5 po d6)

(moveto d4 pa po)

(movefrom d4 po d5)

(moveto d1 d2 po)

(movefrom d1 po d4)

(moveto d2 d3 po)

(movefrom d2 po pa)

(moveto d3 pb po)

(moveto d2 pa d3)

(moveto d1 d4 d2)

(movefrom d1 d2 pb)

(movefrom d2 d3 pa)

(movefrom d3 po d4)

(moveto d2 pa po)

(movefrom d2 po d3)

(moveto d1 pb po)

(movefrom d1 po d2)

1. **Bonus: The textbook gives an example problem in the Air Cargo Domain. Implement the problem domain in PDDL such that it can solve the following instance problem (10 points).**

Ans: Implemented and shared as Air-Cargo\_domain.pddl

Code:

;; STRIPS domain of the Air-Cargo

(define (domain air-cargo)

(:requirements :strips)

(:predicates

(At ?c ?a) ; c is at airport

(In ?c ?p) ; c is in plane p

(Cargo ?c) ; c is a cargo

(Airport ?a) ; a is an airpot

(Plane ?p) ; p is a plane

)

; Load cargo to plane

(:action LOAD

:parameters (?cargo ?plane ?airport)

:precondition (and (At ?cargo ?airport) (At ?plane ?airport) (Cargo ?cargo) (Plane ?plane) (Airport ?airport))

:effect (and (not (At ?cargo ?airport)) (In ?cargo ?plane)))

;unload cargo from plane

(:action UNLOAD

:parameters (?cargo ?plane ?airport)

:precondition (and (In ?cargo ?plane) (At ?plane ?airport) (Cargo ?cargo) (Plane ?plane) (Airport ?airport))

:effect (and (At ?cargo ?airport) (not (In ?cargo ?plane))))

; Fly from one airport to another

(:action FLY

:parameters (?plane ?from ?to)

:precondition (and (At ?plane ?from) (Plane ?plane) (Airport ?from) (Airport ?to))

:effect (and (not (At ?plane ?from)) (At ?plane ?to)))

)

REPORT:

--- OK.

Match tree built with 20 nodes.

PDDL problem description loaded:

Domain: AIR-CARGO

Problem: PB1

#Actions: 20

#Fluents: 12

Landmarks found: 2

Starting search with IW (time budget is 60 secs)...

rel\_plan size: 6

#RP\_fluents 6

Caption

{#goals, #UNnachieved, #Achieved} -> IW(max\_w)

{2/2/0}:IW(1) -> [2][3][4]rel\_plan size: 3

#RP\_fluents 3

{2/1/1}:IW(1) -> [2][3][4]rel\_plan size: 0

#RP\_fluents 0Plan found with cost: 6

Total time: 4.47035e-10

Nodes generated during search: 29

Nodes expanded during search: 20

IW search completed

Path:

(load c2 p2 jfk)

(fly p2 jfk sfo)

(unload c2 p2 sfo)

(load c1 p2 sfo)

(fly p2 sfo jfk)

(unload c1 p2 jfk)