Self-Check 10

Problem 1: Permissible Actions

Action Definition:

(fly ?plane ?from ?to)

Precondition: (plane ?plane), (airport ?from), (airport ?to), (at ?plane ?from)

Add List: (at ?plane ?to)

Delete List: (at ?plane ?from)

Current State:

(plane 1973)

(airport SFO)

(airport JFK)

(at 1973 SFO)

Objective: Demonstrate that the fly action is permissible in the current state. Develop an evaluation strategy and work it out step by step.

Evaluation Strategy:

To determine if an action is permissible (applicable) in a given state, we need to:

1. Identify all variables in the action and possible bindings from the current state.
2. Match the preconditions of the action with facts in the current state using variable bindings.
3. Ensure all preconditions are satisfied with consistent variable assignments.
4. Confirm that the action can be applied.

Step-by-Step Solution:

1. Identify Variables:
    - ?plane
    - ?from

- ?to

2. Possible Bindings from Current State:
    - Planes:
        - ?plane can be 1973 (since (plane 1973) is in the current state).
    - Airports:
        - ?from and ?to can be SFO or JFK (since (airport SFO) and (airport JFK) are in the current state).
3. Match Preconditions:
    - (plane ?plane):
        - With ?plane = 1973, (plane 1973) is satisfied.
    - (airport ?from):
        - Possible ?from values: SFO, JFK.
    - (airport ?to):
        - Possible ?to values: SFO, JFK.
    - (at ?plane ?from):
        - (at 1973 ?from) must be in the current state.
        - From the current state, (at 1973 SFO) exists, so ?from = SFO.
4. Assign Variable Values:
    - ?plane = 1973
    - ?from = SFO
    - ?to = JFK (since we need to fly to a different airport)
5. Verify All Preconditions:
    - (plane 1973): True
    - (airport SFO): True
    - (airport JFK): True
    - (at 1973 SFO): True

6. Conclusion:
    - All preconditions are satisfied with the variable assignments:
        - ?plane = 1973
        - ?from = SFO
        - ?to = JFK
    - Therefore, the fly action is permissible in the current state.

Answer:

By systematically matching the action's preconditions with facts in the current state and assigning variables accordingly, we demonstrated that the fly action is permissible with the assignment (fly 1973 SFO JFK).

Problem 2: A Multiplicity of Permissibility

Extended Current State:

(plane 1973)

(plane 2749)

(plane 97)

(plane 1211)

(airport SFO)

(airport JFK)

(airport ORD)

(at 1973 SFO)

(at 2749 JFK)

(at 97 ORD)

(at 1211 SFO)

Objective: Show all the different ways the fly action is permissible in the current state using the evaluation strategy from Problem 1. Revise the evaluation strategy to account for multiple permissible actions.

Revised Evaluation Strategy:

To handle multiple permissible actions, we need to:

1. Iterate over all possible bindings for the variables ?plane, ?from, and ?to.
2. For each plane:
    - Identify its current location (?from).
    - Consider all other airports as possible destinations (?to), excluding the current location.
3. Check Preconditions for each combination.
4. List all permissible actions with valid variable bindings.

Step-by-Step Solution:

1. List of Planes and Their Locations:

Plane	Current Location (?from)
1973	SFO
2749	JFK
97	ORD
1211	SFO

2. List of Airports (?to options): SFO, JFK, ORD
3. Generate Possible Actions:
    - For Plane 1973 (at SFO):
        - Possible ?to: JFK, ORD
        - Actions:

1. (fly 1973 SFO JFK)
2. (fly 1973 SFO ORD)

- For Plane 2749 (at JFK):
  - Possible ?to: SFO, ORD
  - Actions:
    1. (fly 2749 JFK SFO)
    2. (fly 2749 JFK ORD)
- For Plane 97 (at ORD):
  - Possible ?to: SFO, JFK
  - Actions:
    1. (fly 97 ORD SFO)
    2. (fly 97 ORD JFK)
- For Plane 1211 (at SFO):
  - Possible ?to: JFK, ORD
  - Actions:
    1. (fly 1211 SFO JFK)
    2. (fly 1211 SFO ORD)

4. Check Preconditions for Each Action:
   - All preconditions are satisfied for each action because:
     - (plane ?plane) is true for all planes.
     - (airport ?from) and (airport ?to) are true for all airports.
     - (at ?plane ?from) is true based on the current locations.
     - We exclude actions where ?from = ?to to avoid redundant actions.
5. Conclusion:
   - There are 8 permissible fly actions in the current state.

Revised Evaluation Strategy:

- Account for Multiple Bindings:
    - Generate all combinations of variables that satisfy the preconditions.
- Iterate Over Entities:
    - For each plane, consider all valid destinations.
- Collect All Valid Actions:
    - Store permissible actions in a list or data structure.

Answer:

By expanding the evaluation strategy to consider all combinations of planes and destinations, we identified the following permissible fly actions:

1. (fly 1973 SFO JFK)
2. (fly 1973 SFO ORD)
3. (fly 2749 JFK SFO)
4. (fly 2749 JFK ORD)
5. (fly 97 ORD SFO)
6. (fly 97 ORD JFK)
7. (fly 1211 SFO JFK)
8. (fly 1211 SFO ORD)

Problem 3: Forward Planning

Updated Action Definitions:

1. Fly Action:
2. (fly ?plane ?from ?to)
3. Precondition: (plane ?plane), (airport ?from), (airport ?to), (at ?plane ?from), (fueled ?plane)
4. Add List: (at ?plane ?to), (unfueled ?plane)

5. Delete List: (at ?plane ?from), (fueled ?plane)
6. Fuel Action:
7. (fuel ?plane)
8. Precondition: (plane ?plane), (unfueled ?plane)
9. Add List: (fueled ?plane)
10. Delete List: (unfueled ?plane)

Current State:

(plane 1973)

(plane 2749)

(plane 97)

(plane 1211)

(airport SFO)

(airport JFK)

(airport ORD)

(at 1973 SFO)

(at 2749 JFK)

(at 97 ORD)

(at 1211 SFO)

(fueled 1973)

(unfueled 2749)

(unfueled 97)

(fueled 1211)

Goal State:

(at 1973 ORD)

(at 1211 JFK)

(at 2749 SFO)

Objective: Use a forward planning algorithm to find a plan that achieves the goal state. Work out the solution in detail.

Planning Approach:

1. Identify the differences between the current state and the goal state.
2. Determine the necessary actions to transition from the current state to the goal state.
3. Sequence the actions while respecting preconditions and effects.
4. Consider resource constraints (e.g., fuel status).

Step-by-Step Solution:

1. Analyze Each Plane Individually:

Plane 1973:

- Current Location: SFO
- Goal Location: ORD
- Fuel Status: Fueled

Action Needed: Fly from SFO to ORD.

Check Preconditions for (fly 1973 SFO ORD):

- (plane 1973): True
- (airport SFO): True
- (airport ORD): True
- (at 1973 SFO): True
- (fueled 1973): True

Action is Permissible.

Effects:

- Add: (at 1973 ORD), (unfueled 1973)

- Delete: (at 1973 SFO), (fueled 1973)

Plane 1211:

- Current Location: SFO
- Goal Location: JFK
- Fuel Status: Fueled

Action Needed: Fly from SFO to JFK.

Check Preconditions for (fly 1211 SFO JFK):

- (plane 1211): True
- (airport SFO): True
- (airport JFK): True
- (at 1211 SFO): True
- (fueled 1211): True

Action is Permissible.

Effects:

- Add: (at 1211 JFK), (unfueled 1211)
- Delete: (at 1211 SFO), (fueled 1211)

Plane 2749:

- Current Location: JFK
- Goal Location: SFO
- Fuel Status: Unfueled

Action Needed: Fuel the plane and then fly from JFK to SFO.

First Action: (fuel 2749)

- Preconditions:
    - (plane 2749): True

    - (unfueled 2749): True
- Effects:
    - Add: (fueled 2749)
    - Delete: (unfueled 2749)

Second Action: (fly 2749 JFK SFO)

- Preconditions:
    - (plane 2749): True
    - (airport JFK): True
    - (airport SFO): True
    - (at 2749 JFK): True
    - (fueled 2749): True (achieved after fueling)
- Effects:
    - Add: (at 2749 SFO), (unfueled 2749)
    - Delete: (at 2749 JFK), (fueled 2749)

2. Sequence the Actions:

- Action 1: (fuel 2749)
- Action 2: (fly 2749 JFK SFO)
- Action 3: (fly 1973 SFO ORD)
- Action 4: (fly 1211 SFO JFK)

Note: Actions for planes at SFO can be sequenced independently of fueling 2749, as there are no conflicts.

3. Verify Action Ordering and Preconditions:

- Action 1: (fuel 2749)
    - Preconditions met in the initial state.

- Action 2: (fly 2749 JFK SFO)
    - Preconditions met after Action 1.
- Action 3: (fly 1973 SFO ORD)
    - Preconditions met in the initial state.
- Action 4: (fly 1211 SFO JFK)
    - Preconditions met in the initial state.

Note: Actions 3 and 4 can occur concurrently or before Action 2, as there are no dependencies between them.

4. Final Plan:

1. Fuel Plane 2749:
    - (fuel 2749)
2. Fly Plane 1973 to ORD:
    - (fly 1973 SFO ORD)
3. Fly Plane 1211 to JFK:
    - (fly 1211 SFO JFK)
4. Fly Plane 2749 to SFO:
    - (fly 2749 JFK SFO)

Alternative Sequencing:

- Option 1:
    1. (fly 1973 SFO ORD)
    2. (fly 1211 SFO JFK)
    3. (fuel 2749)
    4. (fly 2749 JFK SFO)
- Option 2:
    1. (fuel 2749)

2. (fly 2749 JFK SFO)
3. (fly 1973 SFO ORD)
4. (fly 1211 SFO JFK)

Conclusion:

- The final plan achieves the goal state.
- The order of independent actions can be flexible.
- All preconditions are met at each step.

Answer:

Using a forward planning algorithm, the plan to achieve the goal state is:

1. Fuel Plane 2749:
    - Apply (fuel 2749) to ensure the plane is fueled.
2. Fly Plane 1973 from SFO to ORD:
    - Apply (fly 1973 SFO ORD).
3. Fly Plane 1211 from SFO to JFK:
    - Apply (fly 1211 SFO JFK).
4. Fly Plane 2749 from JFK to SFO:
    - Apply (fly 2749 JFK SFO).

This sequence of actions transitions the initial state to the goal state while satisfying all action preconditions.

Digging Deeper

1. Adding Capacity Constraints

Objective: Modify the problem to specify that only planes of a specific capacity need to be at certain airports in the goal state.

Solution:

Introduce a New Predicate:

- (capacity ?plane ?size)
    - Represents the capacity of each plane (e.g., small, medium, large).

Example Capacity Assignments:

(capacity 1973 large)

(capacity 2749 medium)

(capacity 97 small)

(capacity 1211 large)

Modify Goal State:

- Specify capacity requirements at airports.
- For example:
    - (at ?plane ORD), (capacity ?plane large)

Adjust Action Preconditions (if necessary):

- Ensure actions consider plane capacities if certain actions are restricted.
- For example, only planes with capacity large can fly to specific airports.

Implementation Steps:

1. Update Plane Definitions:
    - Add capacity information to each plane.
2. Modify Goal State:
    - Include capacity requirements for planes at airports.
3. Adjust Action Definitions:
    - If actions are capacity-dependent, update preconditions accordingly.
    - For example, a new action (fly-large ?plane ?from ?to) with a precondition (capacity ?plane large).

Answer:

By adding the (capacity ?plane ?size) predicate, we can specify capacity constraints in the goal state, ensuring that only planes of a specific capacity are required at certain airports.

2. Combining Forward Planning and A Search Pathfinding

Objective: Describe a problem that combines forward planning with A* search pathfinding.

Problem Description:

Scenario: A delivery drone must deliver packages to various locations in a city. The drone has to:

- Plan High-Level Actions:
    - Pick up packages.
    - Deliver packages.
    - Recharge when battery is low.
- Navigate the Environment:
    - Find optimal paths between locations considering obstacles, no-fly zones, and battery constraints.

Combining Forward Planning and A* Search:

1. Forward Planning:
    - Used to plan the sequence of high-level actions (pickup, delivery, recharge).
    - Considers preconditions and effects of actions.
    - Manages resources (e.g., battery levels, package capacity).
2. A* Search Pathfinding:
    - Used within the move action to find the shortest or safest path between two locations.
    - Considers the physical layout of the environment.
    - Accounts for dynamic obstacles and constraints.

Implementation:

- Action Definition for Move:
- (move ?from ?to)
- Precondition: (at drone ?from), (path ?from ?to)

- Add List: (at drone ?to)
- Delete List: (at drone ?from)
- During the move Action:
    - Invoke A* search to compute the optimal path.
    - Use path cost as part of the planning (e.g., to minimize total travel time).

Benefits:

- Efficient Planning:
    - High-level planning ensures goals are met.
    - A* search optimizes paths, reducing travel time and energy consumption.

Answer:

A delivery drone scenario effectively combines forward planning for high-level action sequencing with A* search for optimal pathfinding between locations. The forward planner handles action order and resource management, while A* search computes efficient routes, resulting in a cohesive and efficient solution.