Unit 13 - Week 11

Assignment 11

The due date for submitting this assignment has passed.
As per our records you have not submitted this assignment.

Due on 2019-10-16, 23:59 IST.

Topics: Planning: Forward/Backward Search, Goal Stack Planning, Sussman's Anomaly

1) Which of the following collectively define a Planning problem? *1 point*

- Domain Description
- Initial and Goal state descriptions
- Action Specification
- Planning agent

No, the answer is incorrect.
Score: 0
Accepted Answers:
Domain Description
Initial and Goal state descriptions
Action Specification

2) Mark all the true statements. STRIPS is.. *1 point*

- a set of notations that can express an automated planning problem
- a program designed to plan in the domain of blocks world
- one of the oldest planning description languages
- not an automated planner

No, the answer is incorrect.
Score: 0
Accepted Answers:
a set of notations that can express an automated planning problem
a program designed to plan in the domain of blocks world
one of the oldest planning description languages

3) Mark all the true statements. PDDL is.. *1 point*

- a set of notations that can express an automated planning problem
- a planning domain description language
- a backward state space planner
- a forward state space planner

No, the answer is incorrect.
Score: 0
Accepted Answers:
a set of notations that can express an automated planning problem
a planning domain description language

4) *Progression* operation is _____ over state space where as *Regression* operation is _____ over state space *1 point*

- Not closed, closed
- complete, sound
- closed, not closed
- sound, complete

No, the answer is incorrect.
Score: 0
Accepted Answers:
closed, not closed

5) Which of the following is/are true regarding Sussman's Anomaly? *1 point*

- It demonstrates that subgoals already achieved may be undone later by a linear planning algorithm
- It never occurs when subgoals are considered in sequence
- It occurs when subgoals may not be serializable
- It doesn't occur in Blocks world domain

No, the answer is incorrect.
Score: 0
Accepted Answers:
It demonstrates that subgoals already achieved may be undone later by a linear planning algorithm
It occurs when subgoals may not be serializable

6) We say that a state S *is progressed through the action a to state S'* *1 point*

- $S' = (S - effects^-(a)) \cap effects^+(a)$
- $S = (S - effects^-(a)) \cap effects^+(a)$
- $S' = (S - effects^-(a)) \cup effects^+(a)$
- $S = (S - effects^-(a)) \cup effects^+(a)$

No, the answer is incorrect.
Score: 0
Accepted Answers:
$S' = (S - effects^-(a)) \cup effects^+(a)$

7) When is an action *a relevant* to a goal *g* ? *1 point*

- $(g \cap effects^+(a) \neq \phi) \wedge (g \cap effects^-(a) = \phi)$
- $(g \cap effects^+(a) = \phi) \wedge (g \cap effects^-(a) \neq \phi)$
- $(g \cap effects^+(a) \neq \phi) \wedge (g \cap effects^-(a) \neq \phi)$
- $(g \cap effects^+(a) = \phi) \wedge (g \cap effects^-(a) = \phi)$

No, the answer is incorrect.
Score: 0
Accepted Answers:
$(g \cap effects^+(a) \neq \phi) \wedge (g \cap effects^-(a) = \phi)$

8) When is an action a *applicable* on a state S? *1 point*

- preconditions(a) $\cap$ S $\neq \phi$
- preconditions(a) $\cap$ S $= \phi$
- preconditions(a) $\subset$ S
- preconditions(a) - $effects^-(a) \subset S$

No, the answer is incorrect.
Score: 0
Accepted Answers:
preconditions(a) $\subset$ S

9) Given a goal *g* and a relevant action a, we say *g* has regressed to the goal *g'* *1 point*

- $g' = (g - effects^+(a)) \cup preconditions(a)$
- $g' = (g - effects^-(a)) \cup preconditions(a)$
- $g' = (g - effects^+(a)) \cap preconditions(a)$
- $g' = (g - effects^-(a)) \cap preconditions(a)$

No, the answer is incorrect.
Score: 0
Accepted Answers:
$g' = (g - effects^+(a)) \cup preconditions(a)$

10) Which of the following is/are true about Goal Stack Planning *1 point*

- The algorithm is complete
- It combines informed search of BSSP with the soundness of FSSP
- It can always solve simple blocks world planning problems
- It follows the approach of linear planning
- It uses two stacks, one for storing subgoals and the other for operators

No, the answer is incorrect.
Score: 0
Accepted Answers:
It combines informed search of BSSP with the soundness of FSSP
It follows the approach of linear planning

11) Given below are the conditions that are checked when the goal stack is popped in Goal stack planning algorithm. Match them with the correct steps given in the option: *2 points*

1. If stack top is a compound goal...
2. If stack top is a satisfied goal...
3. If stack top is a single unsatisfied goal...
4. If stack top is an action...

a. ...just pop it from the stack.
b. ...pop it from the stack, add it to the plan and change the state description based on the effects of the action.
c. ...then push its unsatisfied subgoals on the stack.
d. ...replace it by an action and push the action's precondition on the stack to satisfy the condition

- 1-a, 2-b, 3-c, 4-d
- 1-c, 2-a, 3-d, 4-b
- 1-d, 2-a, 3-d, 4-b
- 1-c, 2-d, 3-a, 4-b

No, the answer is incorrect.
Score: 0
Accepted Answers:
1-c, 2-a, 3-d, 4-b

Given the following planning operators:

Pickup(X)
Preconditions: ontable (X) ∧ clear (X) ∧ armempty
Add: holding (X)
Delete: ontable (X) ∧ armempty

Putdown(X)
Preconditions: holding (X)
Add: ontable (X) ∧ armempty
Delete: holding (X)

Unstack(X,Y)
Preconditions: on (X, Y) ∧ clear (X) ∧ armempty
Add: holding (X) ∧ clear (Y)
Delete: on (X, Y) ∧ armempty

Stack(X,Y)
Preconditions: holding (X) ∧ clear (Y)
Add: on (X, Y) ∧ clear (X) ∧ armempty
Delete: holding (X) ∧ clear (Y)

Assume the following ABBREVIATIONS for actions in the blocks world (to make your answers shorter). Here X and Y are variables that can be A or B or C for this problem.

PiX = Pickup(X), PuX = Putdown(X), SXY = Stack(X,Y), UXY = Unstack(X,Y)

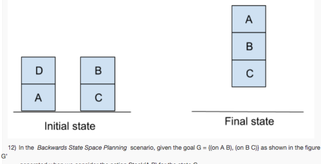
tX = onTable(X), cX = clear(X), hX = holding(X), oXY = on(X,Y)

AE = Arm Empty

Given a planning problem depicted in the figure below, the Initial and Goal states are,

Initial State: onTable(A), onTable(C), clear(D), clear(B), on(D,A), on(B,C), AE
Goal Description: on(A,B), on(B,C)

Or in abbreviated notation
Initial State: tA, tC, cD, cB, oDA, oBC, AE
Goal Description: oAB, oBC



12) In the *Backwards State Space Planning* scenario, given the goal G = {(on A B), (on B C)} as shown in the figure above, what would be the subgoal G' generated when we consider the action Stack(A,B) for the state G *2 points*

- G' = {(on B C), (holding A), (clear B)}
- G' = {(holding A), (clear B) }
- G' = {(holding A), (clear B), ~ArmEmpty }
- The action Stack(A,B) is not relevant for G

No, the answer is incorrect.
Score: 0
Accepted Answers:
G' = {on B C), (holding A), (clear B)}

13) List the set of *applicable* actions for the planning problem described above. The actions must be arranged in alphabetical order or LEXICOGRAPHIC order. The following list of comma separated actions (with no blanks) is an illustration - PuC,SAB,UBC

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) UBC,UDA
(Type: String) UBC, UDA

2 points

14) List the set of *relevant* actions for the planning problem described above. The actions must be arranged in alphabetical order. The following list of comma separated actions is an illustration - PuC,SAB,UBC

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) SAB,SBC
(Type: String) SAB, SBC

2 points

15) For the planning problem specified above which of the following would be the first action to be pushed on the stack by Goal stacking planning algorithm *2 points*

- Stack(A,B)
- Stack(B,C)
- Unstack(D,A)
- Unstack(B,C)

No, the answer is incorrect.
Score: 0
Accepted Answers:
Stack(A,B)