

ASSIGNMENT 6

SOLN.

1. Task 1 is a Programming Task

Any file that is able to solve the problem is acceptable.

2. Predicates

$(\text{onleft } x) = x \text{ is on left}$

$(\text{onright } x) = x \text{ is on right.}$

$(\text{ischild } x) = x \text{ is child.}$

$(\text{isboat } x) = x \text{ is boat.}$

$(\text{isadult } x) = x \text{ is adult}$

[This one is not necessary but can be included if you want]

Initial State

$(\text{ischild } c_1) \wedge (\text{ischild } c_2) \wedge (\text{isboat } b) \wedge$
 $(\text{onleft } b) \wedge (\text{onleft } c_1) \wedge (\text{onleft } c_2) \wedge$
 $(\text{onleft } a_1) \wedge (\text{onleft } a_2)$

Goal State

$(\text{onright } a_1) \wedge (\text{onright } a_2) \wedge (\text{onright } c_1) \wedge$
 $(\text{onright } c_2)$

Operations

ACTION: $\text{ONERORLIFT}(x, y)$

PRECOND: $(\text{onleft } x) \wedge (\text{onleft } y) \wedge (\text{isboat } y)$

EFFECT: $(\text{onright } x) \wedge (\text{onright } y) \wedge (\text{not } (\text{onleft } x)) \wedge$
 $(\text{not } (\text{onleft } y))$

ACTION: Two GO RIGHT (x, y, z)

PRECOND: $(\text{onleft } x) \wedge (\text{onleft } y) \wedge (\text{onleft } z) \wedge$
 $(\text{ischild } x) \wedge (\text{ischild } y) \wedge (\text{isboat } z)$

EFFECT: $(\text{onright } x) \wedge (\text{onright } y) \wedge (\text{onright } z) \wedge$
 $(\text{not } (\text{onleft } x)) \wedge (\text{not } (\text{onleft } y)) \wedge (\text{not } (\text{onleft } z))$

ACTION: ONE GO LEFT (x, y)

PRECOND: $(\text{onright } x) \wedge (\text{onright } y) \wedge (\text{isboat } y)$

EFFECTS: $(\text{onleft } x) \wedge (\text{onleft } y) \wedge (\text{not } (\text{onright } x)) \wedge$
 $(\text{not } (\text{onright } y))$

ACTION: TwoGoLEFT (x, y, z)

PRECOND: $(\text{onright } x) \wedge (\text{onright } y) \wedge (\text{onright } z) \wedge$
 $(\text{ischild } x) \wedge (\text{ischild } y) \wedge (\text{isboat } z)$

EFFECT: $(\text{onleft } x) \wedge (\text{onleft } y) \wedge (\text{onright } z) \wedge$
 $(\text{not } (\text{onright } x)) \wedge (\text{not } (\text{onright } y)) \wedge$
 $(\text{not } (\text{onright } z))$ [ans could at end]

TASK 3

Execution Monitoring / online Replanning:

The actions are not changed. The plan is generated as though it is a deterministic world. Only when execty is it monitored and replanning occurs if an action fails.

Conditional Planning.

The following tasks are modified.

ACTION: $\text{ONE_GO_RIGHT}(x, y)$

PRECOND: $(\text{onleft } x) \wedge (\text{onleft } y) \wedge (\text{isboat } y)$

EFFECT: $\left[(\text{onright } x) \wedge (\text{onright } y) \wedge (\text{not } (\text{onleft } x)) \wedge (\text{not } (\text{onleft } y)) \right] \vee \left[(\text{onleft } x) \wedge (\text{onleft } y) \right]$

ACTION: $\text{ONE_GO_LEFT}(x, y)$

PRECOND: $(\text{onright } x) \wedge (\text{onright } y) \wedge (\text{isboat } y)$

EFFECTS: $\left[(\text{onleft } x) \wedge (\text{onleft } y) \wedge (\text{not } (\text{onright } x)) \wedge (\text{not } (\text{onright } y)) \right] \vee \left[(\text{onright } x) \wedge (\text{onright } y) \right]$

Task 4

Preconditions of $aaa(B, c)$ are in S1.

So this action is applicable. Applying the actions results in.

(A Ht1)

(B Ht1)

(C Ht1)

(ppp1 B C)

(ppp2 A)

~~(ppp2 B)~~

(ppp3 C)

(eee1 A C)

~~(eee2 C)~~

~~(eee3 C)~~

(eee3 A)

(eee1 B C)

(eee2 B)

Task 5

There are 4 predicates which can take
[1 3] arguments.

So total number of ways 5 constants
can be assigned to it is. $[4 \times 5^1 \quad 4 \times 5^3]$
 $= [20 \quad 500]$ possible predicate assignment.

For n predicates, You can have states
where none of them are true to states
where all of them are true

$$nC_0 + nC_1 + \dots + nC_n = \sum_{i=0}^n nC_i$$

So in our case the tight bound on
number of states is.

$$\left[\sum_{i=0}^{20} {}^{20}C_i \quad \sum_{i=0}^{500} {}^{500}C_i \right]$$
$$= \left[2^{20} \quad 2^{500} \right]$$

TASK 2 [ans contd.]

This is one possible plan to solve the problem.

TWO GO RIGHT (c_1, c_2, b)

ONE GO LEFT (c_1, b)

ONE GO RIGHT (a_1, b)

ONE GO LEFT (c_2, b)

TWO GO RIGHT (c_1, c_2, b)

ONE GO LEFT (c_1, b)

ONE GO RIGHT (a_2, b)

ONE GO LEFT (c_2, b)

TWO GO RIGHT (c_1, c_2, b)