AI Planning for Autonomy

Problem Set VI: Relaxed Plan Heuristic and Iterated Width

1. In a blocks-world problem, the agent's aim is to stack the blocks as in Figure 1.

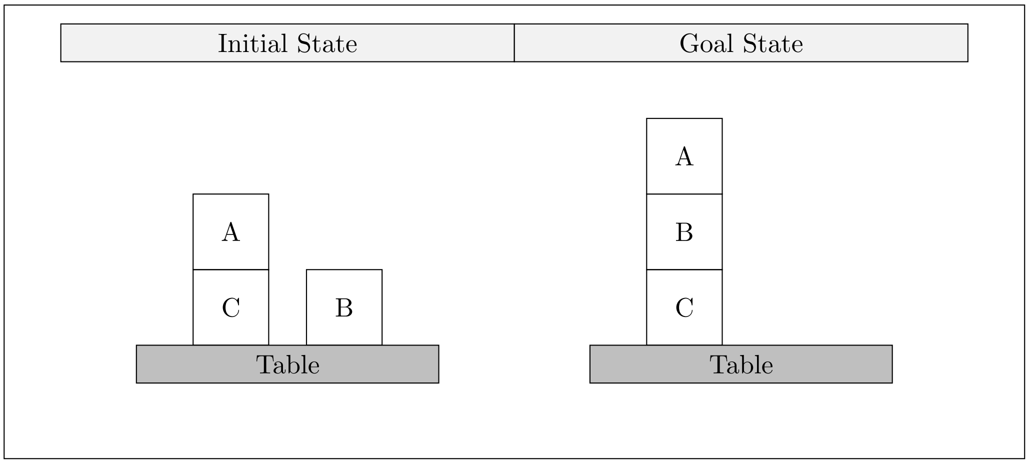

Figure 1: An Initial (Left hand side) and Goal (Right hand side) state of a blocks-world problem.

Compute the values of each of the following heuristics for this problem

- h^{ff}: Use h^{max} for the best-supporters function.
- h^{ff}: Use h^{add} for the best-supporters function.

2. Given the Blocks-World domain, with the initial state where blocks A B, *and* C are on the table, and goal A *on* B:

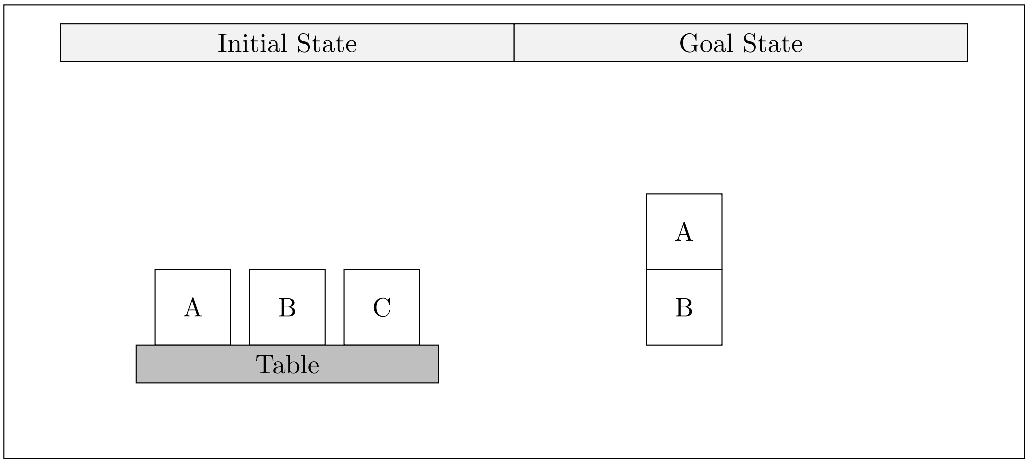

Figure 2: An Initial (Left hand side) and Goal (Right hand side) state of a blocks-world problem.

- Show the IW(1) search tree for this problem, highlighting each state why it passes the novelty pruning test or why is being pruned. IW(1) should solve this problem. Stop as soon as you find a state the satisfies the goal condition.

- Can you think of an initial situation where IW(1) cannot find a solution for the goal on(A,B), but IW(2) does, explain your answer?

  Methodology: Draw a table for each fluent tuple, i.e subset of fluents of size k, where k is the bound used, and progress the search, marking for each node of Breadth First Search which tuples were made true for the first time. (Initialize the table with the initial state)

Relaxed plan = {}
a set of action

open = {on(A,B), on(B,C)}

close = {}

I = {on(A,C), oT(B), oT(C), cl(A), cl(B), aE}

Step 1: on(A,B) → stack(A,B) → pre = {cl(B), h(A)}

Relaxed plan ⇒ Rp = {stack(A,B)}

close = {on(A,B)} open = {on(B,C), h(A)}

h(A) 没有在里面

Step 2: h(A) → unstack(A,C) → pre = {on(A,C), cl(A), aE}

1 unstack(A,B) → pre = {on(A,B), cl(A), aE}

choose the smallest one

3 pickup(A) → pre = {oT(A), cl(A), aE}

3

Rp = {stack(A,B), unstack(A,C)}

close = {on(A,B), h(A)}

open = {on(B,C)}.

---

step 3

on(B,C) → stack(B,C) → pre = {cl(C), h(B)}

Rp = {stack(A,B), unstack(A,C), stack(B,C)}

close = {on(A,B), h(A), on(B,C)}

open = {cl(C), h(B)} ⇒ the element is not in initial and closed lst.

Step 4

h(B) → | unstack(B,C) → pre={on(B,C), cl(B), aE}
unstack(B,A) → pre={on(B,A), cl(B), aE}
pickup(B) → pre={ot(B), cl(B), aE}

Rp={pickup(B), unstack(A,C), stack(B,C), stanck(A,B)}
close={—, h(B)}
open={cl(C)}

cl(C) unstack(B,C)
1 ← unstack(A,C) → 已经在Rp中可以不加入
putdown(C)
stack(C,A)
stack(C,B)