

HW2: Classical Planning and Constraint Satisfaction

CS6300: Artificial Intelligence, Spring 2018
University of Utah

Tucker Hermans

1 Classical Planning

1. Define the preconditions and postconditions (add and delete lists) for the two actions. Assume the agent can only pick up a single object with nothing on it, that it can only hold a single object at a time, and that it can only place the object on a horizontally-flat, open surface.

We will introduce two new fluents. The first $flat(x)$ is true if x is a horizontally-flat box. The second $in_hand(x)$ is true if x is in the agent's hand.

We will use $clear(hand)$ to indicate the agent's hand is empty. Otherwise $clear(x)$ indicates there is nothing on top of x but it is not necessarily flat. Finally, $on(x, y)$ will be used to indicate object x is on object y .

Using these, we can define the preconditions, add list, and delete list for $pick_up(x, y)$ and $place(x, y)$. Note I expanded the definition of $pick_up(x)$ to $pick_up(x, y)$ to specify the object under x .

PC	D	A
$clear(x)$	$on(x, y)$	$clear(y)$
$clear(hand)$	$clear(hand)$	$in_hand(x)$
$on(x, y)$	$clear(x)$	

Table 1: Preconditions, add list, and delete list for $pick_up(x, y)$.

PC	D	A
$in_hand(x)$	$in_hand(x)$	$clear(hand)$
$clear(y)$	$clear(y)$	$clear(x)$
$flat(y)$		$clear(table)$

Table 2: Preconditions, add list, and delete list for $place(x, y)$.

2. How would you express a goal that there are two towers, each of height 3 blocks, with pyramids on top of each tower?

The goal is defined as a conjunction of fluents (broken into two lines for readability):

$$\begin{aligned} & on(A, Table) \wedge on(B, A) \wedge on(C, B) \wedge \neg flat(C) \\ & \wedge on(D, Table) \wedge on(E, D) \wedge on(F, E) \wedge \neg flat(F) \end{aligned}$$

Where A-F are blocks. We don't have to be concerned with the peaks of the towers, C and F, having blocks on them (and increasing the height pass the goal height) as it's an invalid move to *place* or *move* a block onto a block that isn't horizontally flat.

3. Suppose we introduce a new type of *plank* object into the environment, which is long and skinny and must be supported by either the table or two blocks. What necessary changes would you have to make to your planning description to accommodate these changes?

2 Crossword Puzzles

1. Formulate this problem as a CSP where the variables are words. List all the variables and constraints.
2. Formulate this problem as a CSP where the variables are letters. List all the variables and constraints.
3. Suppose we wish to make sure the the crossword puzzle we generate doesn't contain multiple words that are "too similar." We'll say that words w_1 and w_2 are too similar if *either* one can be obtained from the other by changing exactly one character *or* one is a substring of the other. For instance "dog" and "dogs" are too similar (by the second constraint) as are "dog" and "dog" (again, by the second constraint). Similarly, "cats" and "cots" are too similar (by the first constraint). How would you specify these additional constraints in both the by-word and by-letter formulations?