Christina Ringwald, 2571692
Anna Krasilnikova, 2562668
Mirza Misbah Mubeen Baig, 2571567

Exercise 1.

See the rest of archive

Exercise 2

(i) Translate the planning task to STRIPS by defining the facts, actions, initial state and goal.

Facts: $\{L_i : i \in \{None\} \cup Pumpkins\} \cup \{A_c : c \in CarveTools\} \cup$
$\{D_{p,c} : p \in Pumpkins, c \in CarveTools\} \cup$
$\{H_{h,x} : h \in Hands, x \in \{None\} \cup CarveTools\}$

Actions: {*look_at(i), pick_up(h, c), drop(c, h), carve(p, c, h)*}

look_at(i,k) for *i* , *k* in *Pumpkins* or {*None*}
pre : $\{L_k\}$
add : $\{L_i\}$
del : $\{L_k\}$

pick_up(h,c) for *h* in *Hands, c* in *CarveTools*
pre : $\{A_c , H_{h, none}\}$
add : $\{H_{h,c}\}$
del : $\{A_c , H_{h, none}\}$

drop(c,h) for *c* in *CarveTools, h* in *Hands*
pre : $\{H_{h,c}\}$
add : $\{A_c , H_{h, none}\}$
del : $\{H_{h,c}\}$

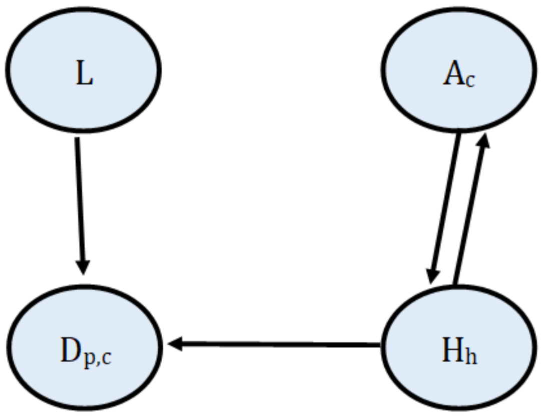
put_stick(p,c,h) for *p* in *Pumpkins, c* in *CarveTools, h* in *Hands*
pre : $\{H_{h,c}, L_p\}$
add : $\{D_{p,c}\}$
del : {}

Initial State: $\{L_{None} , H_{left, None} , H_{right, None} , A_{smiley} , A_{cat} , A_{bat}\}$

Goal: $\{ D_{Muskat,smiley}, D_{Muskat,cat} , D_{Hokkaido,bat}\}$

(ii) Draw the Causal Graph of the planning task.

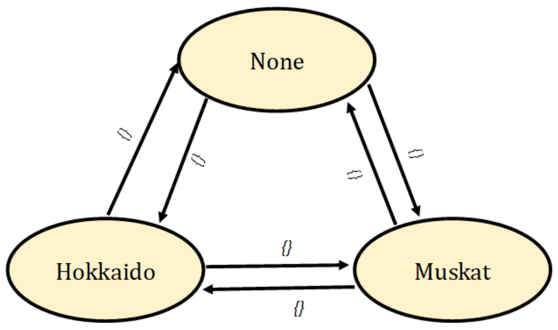
Causal Graph:


$$\begin{tikzcd} L \arrow[d] & A_c \arrow[d, shift right] \\ D_{p,c} & H_h \arrow[l] \arrow[u, shift right] \end{tikzcd}$$

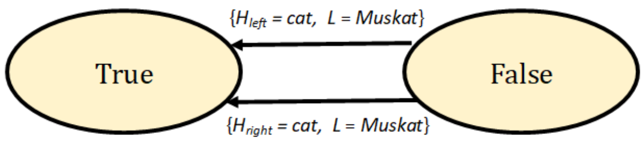
(iii) Draw the DTG for Variables L and $D_{Muskat,cat}$. Annotate the transitions with their (outside) conditions. Which of those transitions are invertible according to the definition given in the lecture (Chapter 5)?

Christina Ringwald, 2571692
Anna Krasilnikova, 2562668
Mirza Misbah Mubeen Baig, 2571567

DTG
L:



$D_{Muskat,cat}$:


True
False
$\{H_{left} = cat,\ L = Muskat\}$
$\{H_{right} = cat,\ L = Muskat\}$

The value transition for L is invertible.

(iv) Explain which task simplification techniques from Chapter 5 can be applied in this planning task. Explain for which (sub-)problem(s) do we need to run a search algorithm. Give a plan for the simplified task and explain how will you recover the solution plan for the original task afterwards?

We can use the Invertible Root Variable simplification. Removing L, we find a plan for the simplified task:

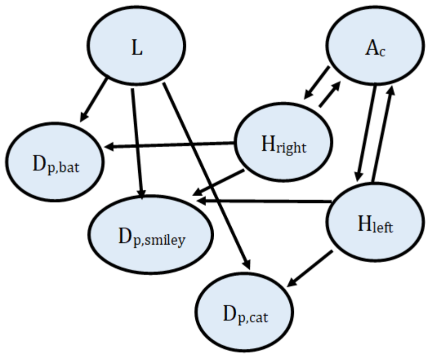
pick_up(*left,smiley*)
pick_up(*right,cat*)
put_stick(*Muskat,smiley,left*)
put_stick(*Muskat,cat,right*)
drop(*left,smiley*)
pick_up(*left,bat*)
put_stick(*Hokkaido,bat,left*)

Then we extend the plan by adding actions involving L in order to achieve the preconditions on L.

pick_up(*right,smiley*)
pick_up(*left,cat*)
look_at(Muskat)
put_stick(*Muskat,smiley,right*)
put_stick(*Muskat,cat,left*)
drop(*right,smiley*)
pick_up(*right,bat*)
look_at(Hokkaido)
put_stick(*Hokkaido,bat,right*)

Christina Ringwald, 2571692
Anna Krasilnikova, 2562668
Mirza Misbah Mubeen Baig, 2571567

(v) Suppose that we pre-assign what to carve to the hands so that we only allow the left hand to carve the cat and the right hand the bat. Re-draw the causal graph. How does this affect to your answer above in part (iv)? Justify your answer.



This does not change the answer to part (iv) because L is still an invertible root node.

Exercise 4 – *Prove that PlanOpt for STRIPS is PSPACE-hard.*

Let's start with considering PlanSAT.
Since any Turing Machine transition from one state to another can be mapped into an operator, then the total number of such operators is proportional to the number of transition times the number of tape squares. Hence, the PSPACE Turing Machine corresponds to a polynomial number of operators[1].
Now we can prove by reduction that PlanOpt is PSPACE hard from PlanSAT.

$PLANSAT(C, \pi)$ input:
planning instance *P* in *C*
values for the parameters in π
Output:

$$\begin{cases} yes, if\ P\ has\ a\ plan \\ no, otherwise \end{cases}$$

$PLANOPT(C, \pi)$:
Input: the same
Output:

$$\begin{cases} yes, if\ P\ has\ a\ shortest\ plan \\ no, otherwise \end{cases}$$

So we can prove PSPACE-hardness from SAT by reduction, or by generic reduction from TM acceptance.

[1] "The Computational Complexity of Propositional STRIPS Planning", T. Bylander, 1994