Exercise 1

Consider the following planning task, where a rover needs to take a picture of the surface of Mars.

- $P = \{\textit{at-A}, \textit{at-B}, \textit{at-C}, \textit{cam-ready}, \textit{pic-taken}\}$
- A =
  - $\textit{move-A-B} = \langle pre : \{\textit{at-A}\}, add : \{\textit{at-B}\}, del : \{\textit{at-A}\}\rangle$
  - $\textit{move-B-C} = \langle pre : \{\textit{at-B}\}, add : \{\textit{at-C}\}, del : \{\textit{at-B}\}\rangle$
  - $\textit{calibrate} = \langle pre : \{\}, add : \{\textit{cam-ready}\}, del : \{\}\rangle$
  - $\textit{take-pic} = \langle pre : \{\textit{cam-ready}, \textit{at-C}\}, add : \{\textit{pic-taken}\}, del : \{\}\rangle$
- $I = \{\textit{at-A}\}$.
- $G = \{\textit{pic-taken}\}$.

i) Construct the formula $\phi^{seq}_{\Pi,1}$

ii) Which pairs of actions have interference in $\forall$-step semantics? Why?

iii) What is the minimum value of L for which the formula $\phi^{seq}_{\Pi,L}$ is satisfiable?

iv) What is the minimum value of L for which the formula $\phi^{\forall\text{-}step}_{\Pi,L}$ is satisfiable?

Solution:

i)

$$
\begin{aligned}
\phi^{seq}_{\Pi,1} = {} & \textit{at-A}^0 \wedge \neg \textit{at-B}^0 \wedge \neg \textit{at-C}^0 \wedge \neg \textit{cam-ready}^0, \neg \textit{pic-taken}^0 \\
& \wedge \textit{pic-taken}^1 \\
& \wedge (\textit{move-A-B}^1 \implies (\textit{at-A}^0 \wedge \textit{at-B}^1 \wedge \neg \textit{at-A}^1)) \\
& \wedge (\textit{move-B-C}^1 \implies (\textit{at-B}^0 \wedge \textit{at-C}^1 \wedge \neg \textit{at-B}^1)) \\
& \wedge (\textit{calibrate}^1 \implies \textit{cam-ready}^1) \\
& \wedge (\textit{take-pic}^1 \implies (\textit{cam-ready}^0 \wedge \textit{at-C}^0 \wedge \textit{pic-taken}^1)) \\
& \wedge ((\textit{at-A}^1 \wedge \neg \textit{at-A}^0) \implies (\bot)) \wedge ((\neg \textit{at-A}^1 \wedge \textit{at-A}^0) \implies (\textit{move-A-B}^1)) \\
& \wedge ((\textit{at-B}^1 \wedge \neg \textit{at-B}^0) \implies (\textit{move-A-B}^1)) \wedge ((\neg \textit{at-B}^1 \wedge \textit{at-B}^0) \implies (\textit{move-B-C}^1)) \\
& \wedge ((\textit{at-C}^1 \wedge \neg \textit{at-C}^0) \implies (\textit{move-B-C}^1)) \wedge ((\neg \textit{at-C}^1 \wedge \textit{at-C}^0) \implies (\bot)) \\
& \wedge ((\textit{cam-ready}^1 \wedge \neg \textit{cam-ready}^0) \implies (\textit{calibrate})) \wedge ((\neg \textit{cam-ready}^1 \wedge \textit{cam-ready}^0) \implies (\bot)) \\
& \wedge ((\textit{pic-taken}^1 \wedge \neg \textit{pic-taken}^0) \implies (\textit{take-pic})) \wedge ((\neg \textit{pic-taken}^1 \wedge \textit{pic-taken}^0) \implies (\bot)) \\
& \wedge \neg(\textit{move-A-B}^1 \wedge \textit{move-B-C}^1) \wedge \neg(\textit{move-A-B}^1 \wedge \textit{calibrate}^1) \wedge \neg(\textit{move-A-B}^1 \wedge \textit{take-pic}^1) \\
& \wedge \neg(\textit{move-B-C}^1 \wedge \textit{calibrate}^1) \wedge \neg(\textit{move-B-C}^1 \wedge \textit{take-pic}^1) \wedge \neg(\textit{calibrate}^1 \wedge \textit{take-pic}^1)
\end{aligned}
$$

ii) move-A-B and move-B-C because they add/delete the same fact (at-B)

iii) 4, which is the length of the optimal solution: move-A-B, move-B-C, calibrate, take-pic

iv) 3, because cam-ready can be done in parallel to move-A-B or move-B-C

Exercise 2

Let Π be a planning task. For each of the following cases, indicate what can you infer about $h^*(I)$ ($i.e., h^*(I) < 5, h^*(I) \leq 5, h^*(I) > 5, h^*(I) \geq 5, h^*(I) = 5, h^*(I) \neq 5$, or other) and justify your answer.

i) $\phi^{\textit{seq}}_{\Pi,5}$ is unsatisfiable

ii) $\phi^{\textit{seq}}_{\Pi,5}$ is satisfiable

iii) $\phi^{\forall\textit{-step}}_{\Pi,5}$ is unsatisfiable

iv) $\phi^{\forall\textit{-step}}_{\Pi,5}$ is satisfiable

Solution:

i) $h^*(I) > 5$, as no solution with 5 steps exists.

ii) $h^*(I) \leq 5$, and the satisfying assignment provides us with the sequence of actions that can be applied (possibly including noop).

iii) $h^*(I) > 5$, as no solution with 5 steps exists (but perhaps a solution with 6 actions in 6 steps exists).

iv) $h^*(I) \leq 5|A|$, (that is, there is a solution applying each action at most five times).

Exercise 3

Table 1 provides the number of BDD nodes in symbolic forward and backward search for a concrete planning task. Considering this data, and considering that symbolic bidirectional search chooses whether to do a forward or backward step by comparing the BDD nodes in the frontier, answer the following questions and justify your answer:

i) How many steps will be required by symbolic bidirectional search to solve the problem?

ii) How many of those steps will be in the forward and how many in the backward direction?

iii) How much time will symbolic bidirectional search take to solve the problem?

iv) What will be the maximum number of nodes in the frontier BDD?

v) What is the length of the plan retrieved by symbolic bidirectional search?

vi) What is the length of the plan retrieved by symbolic forward search?

vii) What is the length of the plan retrieved by symbolic backward search?

Step	Forward		Backward	
	Nodes	Accumulated Time (s)	Nodes	Accumulated Time (s)
1	47	0.50	64	0.49
2	61	0.50	47	0.49
3	71	0.50	47	0.49
4	72	0.50	47	0.49
5	111	0.50	64	0.49
6	172	0.50	91	0.49
7	264	0.50	128	0.49
8	390	0.50	187	0.49
9	476	0.50	240	0.49
10	877	0.50	373	0.49
11	1314	0.50	569	0.49
12	2445	0.51	999	0.49
13	3606	0.52	1524	0.49
14	6698	0.53	2731	0.50
15	9633	0.55	4011	0.51
16	18265	0.58	7352	0.52
17	27102	0.64	11115	0.54
18	53583	0.77	21260	0.59
19	69305	1.00	33219	0.68
20	129679	1.39	62252	0.85
21	144626	2.00	90287	1.14
22	233384	2.85	155914	1.61
23	225292	4.04	194383	2.41
24	304373	5.35	292845	3.52
25	263044	6.90	315181	5.19
26	294367	8.33	398764	7.05
27	232805	9.79	375256	9.47
28	200063	10.8	388079	11.7
29	146009	11.6	327009	14.8
30	88664	12.1	254826	16.9

Table 1: Nodes and accumulated time of symbolic forward and backward search on a blocksworld task with 9 blocks.

Solution:

i) 30, as the number of steps is the same by all 3 algorithms (equal to the optimal plan length h^*)

ii) 14 forward steps and 16 backward steps. It is easy to calculate in this case because the number of nodes in the frontier grows with the steps (until the very end). In the last step, taking the 16th step in the backward direction is preferred over taking the 15th step in the forward direction ($7352 < 9633$).

iii) $0.53 + 0.52 = 1.05s$. Of course, this is an approximation, for those of you curious, these would be the actual result of running symbolic bidirectional search:

Direction	g	Nodes	Accumulated Time (s)
Forward	0	47	0.50
Backward	0	64	0.50
Backward	1	47	0.50
Backward	2	47	0.50
Backward	3	47	0.50
Forward	1	61	0.50
Backward	4	64	0.50
Forward	2	71	0.50
Forward	3	72	0.50
Backward	5	91	0.50
Forward	4	111	0.50
Backward	6	128	0.50
Forward	5	172	0.50
Backward	7	187	0.50
Backward	8	240	0.50
Forward	6	264	0.50
Backward	9	373	0.50
Forward	7	390	0.50
Forward	8	476	0.50
Backward	10	569	0.50
Forward	9	877	0.50
Backward	11	999	0.51
Forward	10	1314	0.51
Backward	12	1524	0.51
Forward	11	2445	0.52
Backward	13	2731	0.52
Forward	12	3606	0.53
Backward	14	4011	0.55
Forward	13	6698	0.56
Backward	15	7352	0.58

iv) 7352, which corresponds to the last step taken by the backward search

v) 30, as the number of steps is the same by all 3 algorithms (equal to the optimal plan length h^*)

vi) 30, as the number of steps is the same by all 3 algorithms (equal to the optimal plan length h^*)

vii) 30, as the number of steps is the same by all 3 algorithms (equal to the optimal plan length h^*)

Exercise 4

Consider the following planning task, where a rover needs to take a picture of the surface of Mars.

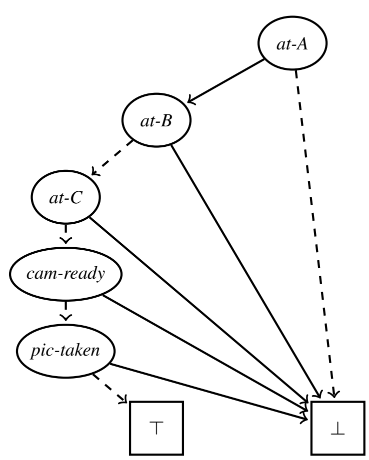
- $P = \{$*at-A*$,$ *at-B*$,$ *at-C*$,$ *cam-ready*$,$ *pic-taken*$\}$
- A =
  - *move-A-B* $= \langle pre : \{$*at-A*$\}, add : \{$*at-B*$\}, del : \{$*at-A*$\}\rangle$
  - *move-B-C* $= \langle pre : \{$*at-B*$\}, add : \{$*at-C*$\}, del : \{$*at-B*$\}\rangle$
  - *calibrate* $= \langle pre : \{\}, add : \{$*cam-ready*$\}, del : \{\}\rangle$
  - *take-pic* $= \langle pre : \{$*cam-ready*$,$ *at-C*$\}, add : \{$*pic-taken*$\}, del : \{\}\rangle$
- $I = \{$*at-A*$\}$.
- $G = \{$*pic-taken*$\}$.

Considering that the BDD variable ordering is $\langle$*at-A*$,$ *at-B*$,$ *at-C*$,$ *cam-ready*$,$ *pic-taken*$\rangle$, draw the following BDDs:

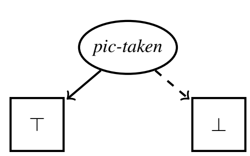
1. I
2. G
3. $TR_{take-pic}$
4. image(I, TR) Hint: TR represents the disjunction of all actions; you don't need to do the operation step by step, simply think which set of states this operation will result in.

Solution:

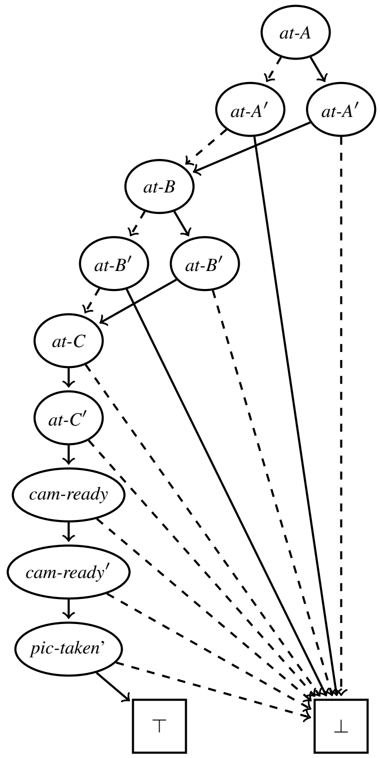
1. I



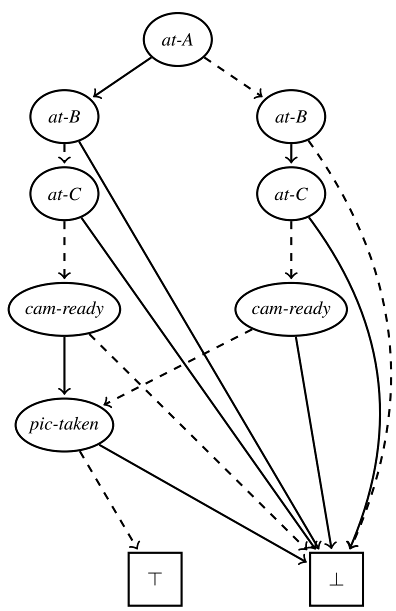
2. G



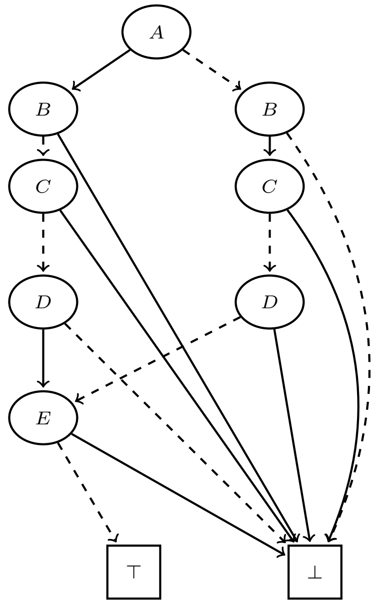
3. $TR_{\textit{take-pic}}$


at-A
at-A′
at-A′
at-B
at-B′
at-B′
at-C
at-C′
cam-ready
cam-ready′
pic-taken’
⊤
⊥

4. *image*(I, TR)


at-A
at-B
at-B
at-C
at-C
cam-ready
cam-ready
pic-taken
⊤
⊥

Exercise 5



Consider the following BDD representing the function f.

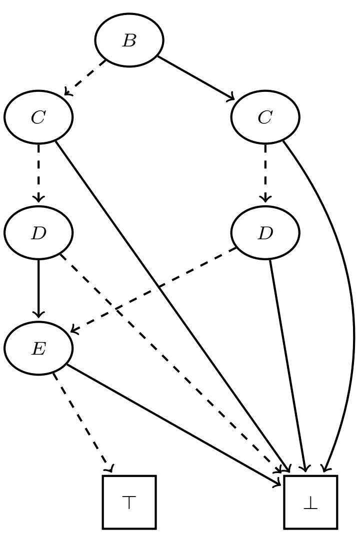
i) Draw the BDD corresponding to $\exists_A f$.

ii) Draw the BDD corresponding to $\exists_{A,B} f$.

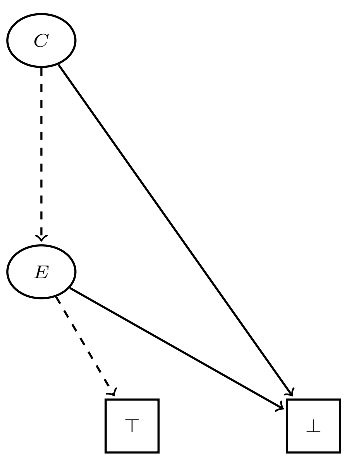
iii) Draw the BDD corresponding to $\exists_{D,E} f$.

iv) In the worst case, if we have an arbitrary BDD and apply existential quantification with respect to k variables, is the size of the resulting BDD polynomial or exponential in the size of the original BDD. Explain why.

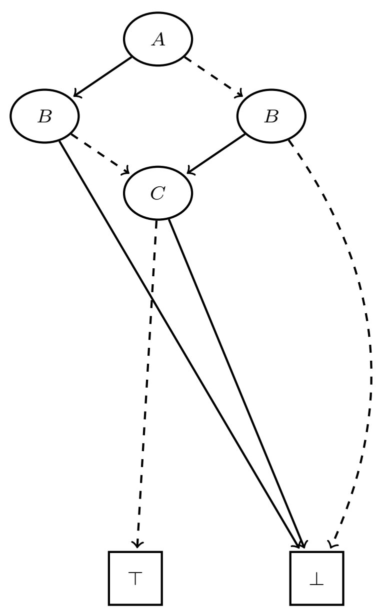
Solution:



(a) i) $\exists_A F$



(b) iii) $\exists_{A,B} F$



(c) iii) $\exists_{D,E} F$

It is exponential even in the restricted case where we abstract the top k variables in the variable ordering. In that case, the results correspond to the disjunction of all nodes in the kth layer. This is a polynomial number of disjunctions (certainly less than the number of nodes in the original BDD), but the size of the resulting BDD may increase exponentially in the number of disjunctions (as we studied in Chapter 9).