You have three containers that may hold 12 liters, 8 liters and 3 liters of water, respectively, as well as access to a water faucet. You can fill a container from the faucet, pour it into another container, or empty it onto the ground. The goal is to measure exactly one liter of water.

1. Give a precise specification of the task as a search problem.
2. Draw the search tree produced by depth-limited search with maximum depth equal to three and elimination of repeated states.

We have 3 variables: A, B, C where $A \in [0, 12]$ $B \in [0, 6]$ $C \in [0, 3]$. The actions are

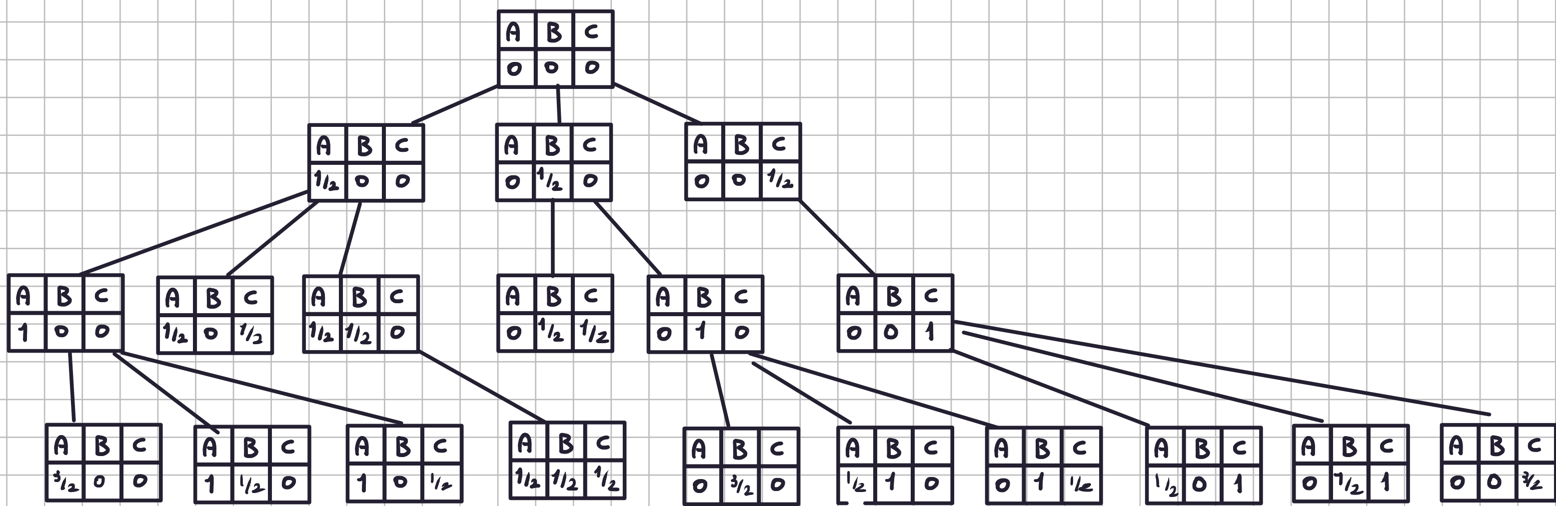
pour: $A \mathrel{+}= \varepsilon$, preconditions: $A \leq 12 - \varepsilon$
$B \mathrel{+}= \varepsilon$, preconditions: $B \leq 6 - \varepsilon$
$C \mathrel{+}= \varepsilon$, preconditions: $C \leq 3 - \varepsilon$

we discretize the water in packet of $\varepsilon > 0$ liters.

exchange: $A \mathrel{+}= \varepsilon$, $B \mathrel{-}= \varepsilon$, prec: $A \leq 12 - \varepsilon$, $B \geq \varepsilon$
$B \mathrel{+}= \varepsilon$ $A \mathrel{-}= \varepsilon$, prec. $B \leq 12 - \varepsilon$, $A \geq \varepsilon$
$\vdots$ $\vdots$
$C \mathrel{+}= \varepsilon$, $B \mathrel{-}= \varepsilon$, prec. $C \leq 3 - \varepsilon$, $B \geq \varepsilon$

empty: $A \mathrel{-}= \varepsilon$, preconditions: $A \geq \varepsilon$
$B \mathrel{-}= \varepsilon$, preconditions: $B \geq \varepsilon$
$C \mathrel{-}= \varepsilon$, preconditions: $C \geq \varepsilon$

For example let $\varepsilon = 1/2$



a) Transform the following formula to CNF, specifying which steps you are applying and giving the intermediate results.

$$(\neg A \rightarrow (B \leftrightarrow C)) \wedge (D \rightarrow \neg(\neg C \rightarrow A))$$

b) For the following formula, use resolution to determine if the formula is satisfiable or unsatisfiable.

$$(\neg A \vee \neg D) \wedge (B \vee C \vee \neg D) \wedge (A \vee \neg B \vee \neg D) \wedge (A \vee B) \wedge (\neg A \vee D) \wedge (A \vee \neg B \vee D) \wedge (\neg C \vee \neg D)$$

c) Perform DPLL with clause learning. Assume that DPLL selects variables in alphabetical order (i.e., A, B, C, D, E,...), and that the splitting rule first attempts the value False (F). Do this until the clause set is proven to be satisfiable or unsatisfiable.

$$\Delta = \{\{A, B, \neg C, D\}, \{B, \neg C, D\}, \{A, \neg D\}, \{\neg A, \neg D\}, \{\neg B, \neg C\}, \{\neg B, C\}, \{A, \neg B, C, D\}, \{C, \neg D, E\}, \{C, \neg D, \neg E\}\}$$

a) $(\neg A \rightarrow (B \leftrightarrow C)) \wedge (D \rightarrow \neg(\neg C \rightarrow A))$ remove $\rightarrow$ and $\leftrightarrow$

$[A \vee ((B \wedge C) \vee (\neg B \wedge \neg C))] \wedge (\neg D \vee \neg(C \vee A))$ de morgan

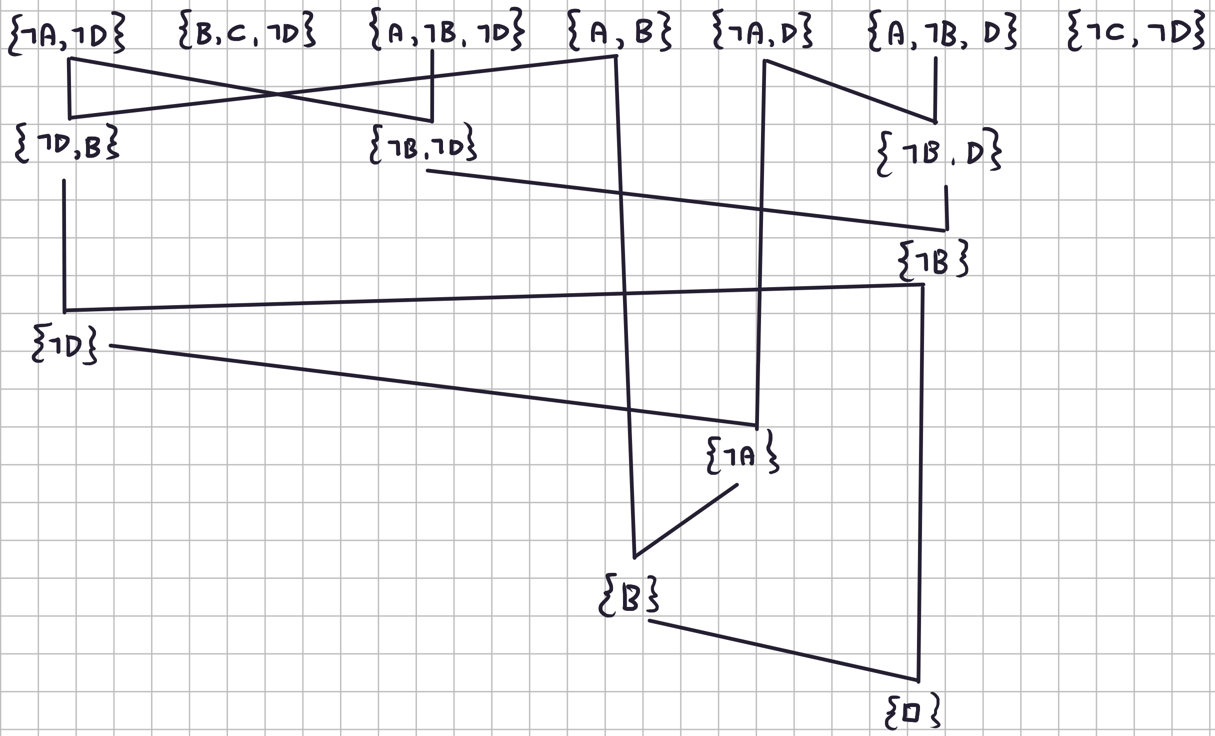
$[A \vee ((B \wedge C) \vee (\neg B \wedge \neg C))] \wedge (\neg D \vee (\neg C \wedge \neg A))$ distribute

$[A \vee (B \wedge C) \vee (\neg B \wedge \neg C)] \wedge [(\neg D \vee \neg C) \wedge (\neg D \vee \neg A)]$ distribute

$[(A \vee B) \wedge (A \vee C) \vee (\neg B \wedge \neg C)] \wedge [(\neg D \vee \neg C) \wedge (\neg D \vee \neg A)]$ distribute

$(A \vee B) \wedge (A \vee C \vee \neg B) \wedge (\neg D \vee \neg C) \wedge (\neg D \vee \neg A)$

b) now we check the satisfiability



$\{\neg A, \neg D\}\ \{A, B\} \Rightarrow \{B, \neg D\}$

$\{\neg A, \neg D\}\ \{A, \neg B, \neg D\} \Rightarrow \{\neg B, \neg D\}$

$\{\neg A, D\}\ \{A, \neg B, D\} \Rightarrow \{\neg B, D\}$

$\left.\begin{matrix}\{\neg B,\neg D\} \\ \{\neg B, D\}\end{matrix}\right\} \{\neg B\}$

$\left.\begin{matrix}\{B, \neg D\} \\ \{\neg B\}\end{matrix}\right\} \{\neg D\}$ and $\{\neg A, D\} \Rightarrow \{\neg A\}$

$\left.\begin{matrix}\{\neg A\} \\ \{A, B\}\end{matrix}\right\} \Rightarrow \{B\}$

$\left.\begin{matrix}\{B\} \\ \{\neg B\}\end{matrix}\right\} \Rightarrow \{\square\}$

c)

$\{A,B,\neg C,D\}\ \{B,\neg C,D\}\ \{A,\neg D\}\ \{\neg A,\neg D\}\ \{\neg B,\neg C\}\ \{\neg B,C\}\ \{A,\neg B,C,D\}\ \{C,\neg D,E\}\ \{C,\neg D,\neg E\}$

choice: A = 0

// $\{B,\neg C,D\}\ \{\neg D\}$ // $\{\neg B,\neg C\}\ \{\neg B,C\}\ \{\neg B,C,D\}\ \{C,\neg D,E\}\ \{C,\neg D,\neg E\}$

UP: D = 0

$\{B,\neg C\}$ // // $\{\neg B,\neg C\}\ \{\neg B,C\}$ // // //

choice: B = 0

$\{\neg C\}$ // //

choice: C = 0 ⇒ Assignment: A:0 B:0 C:0 D:0 E:0

Let $T = \{P, I, G, A\}$ be an STRIPS planning task describing the job of a modern pigeon, that has to go to the place where the message is and take it. Formally, the task is defined as follows:

- P : {*at-pigeon*(x), $x \in \{L_1, L_2, L_3\}$; *mes-found*; *adjacent*(x, y) with $x, y \in \{L_1, L_2, L_3\}$
- I : {*at-pigeon*(L_1)} $\cup$ {*adjacent*(x, y) | $(x, y) \in \{\{L_1, L_2\}, \{L_2, L_3\}, \{L_2, L_1\}, \{L_3, L_2\}\}$}
- G : {*at-pigeon*(L_1), *mes-found*}
- A :
  - fly(x, y):
    - pre: *adjacent*(x, y), *at-pigeon*(x)
    - add: *at-pigeon*(y)
    - del: *at-pigeon*(x)
  - take-message:
    - pre: *at-pigeon*(L_2)
    - add: *mes-found*
    - del:

1. Compute the value $h^{max}(I)$ for the initial state using the dynamic programming algorithm in Ch 20, slide 32 (Please note: that slide is for h^{add}; for h^{max}, you just need to use the max operator in $Cost_i$ instead of the sum). Write down the table of intermediate values for each iteration of the algorithm, until convergence. Use the variable names listed above. See Ch 20, Slide 33 for an example.
2. Compute the value $h^{add}(I)$ for the initial state using the dynamic programming algorithm in Ch 20, slide 32. Write down the table of intermediate values for each iteration of the algorithm, until convergence. Use the variable names listed above. See Ch 20, Slide 34 for an example.

A state is described by: $(atL1, atL2, atL3, msg) \in \{0,1\}^4$, we ignore the constant proposition of the adjacent location. We recall that

$$h^{max}(s,g) = \begin{cases} 0 & \text{if } s \leq g \\ \min_a c(a) + h^1(s, regr(g,a)) & \text{if } |g| = 1 \quad \text{// a regressable} \\ \max_{g' \in g} h^1(s,g') & \text{if } |g| > 1 \end{cases}$$

$$h^{add}(s,g) = \begin{cases} 0 & \text{if } s \leq g \\ \min_a c(a) + h^1(s, regr(g,a)) & \text{if } |g| = 1 \quad \text{// a regressable} \\ \sum_{g' \in g} h^1(s,g') & \text{if } |g| > 1 \end{cases}$$

So, for h^{max} we initialize $T^0(g)$ for $g \in P, |g| = 1$ in state $I = \{atL1\}$

	L1	L2	L3	msg
T^0	0	∞	∞	∞

L1 is regressable on L2, so $T^1(L1) = \min\{T^0(L1), \underbrace{1}_{\text{cost action}} + \underbrace{T^0(L2)}_{\text{regressed}}\} = \min\{0, \infty\} = 0$

L2 is reg. on L1 and L3 : $T^1(L2) = \min\{T^0(L2), 1 + T^0(L1), 1 + T^0(L3)\} = 1$

L3 is reg. on L2, so : $T^1(L3) = \min\{T^0(L3), 1 + T^0(L2)\} = \infty$

msg is reg. on L2 so $T^1(msg) = \dots = \infty$

	L1	L2	L3	msg
T^0	0	∞	∞	∞
T^1	0	1	∞	∞
T^2	0	1	2	2
T^3	0	1	2	2

← we continue the same reasoning and get this table.

$\Rightarrow h^{max}(I, G) = 2$

Now we consider h^{add}, the table is identical, we just have $h^{add}(I,G) = T^3(msg) + T^3(L1) = 2$