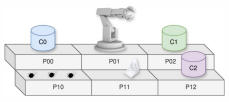
The robotic arm shown in the figure needs to pile some cylinders up in the location P10. More specifically, it must pile them up in the following order: C2 on top of C0, C0 on top of C1, and C1 on the location P10. However, P10 is dirty and must be first cleaned by using the tissue that is located in location P11. After cleaning P10, the tissue must return to its place. Constraints: the robot can only hold one item at a time (either a cylinder or the tissue) and can only pick a cylinder if no other cylinder is on top of it.



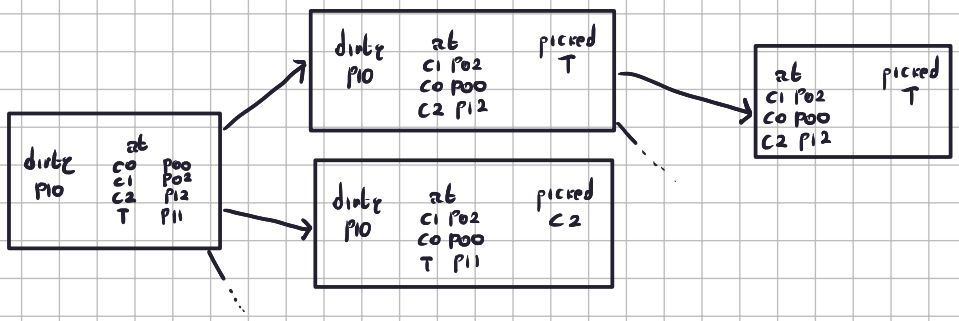
a) Model the problem in PDDL (Planning Domain Definition Language) and define the domain.
b) Based on the domain file in point a), define the problem.
c) Show one possible sequence of actions (plan) leading the robot from the initial state to the goal state.
d) How would you model this problem to apply local search? Which local search algorithms would you apply and which not? Do you think local search would be particularly suitable for this problem? Motivate each of your answers.

```
(:requirments :adl)

(:types     item
            block tissue - item
            loc )

(:predicates  (picked ?I - item)
              (dirty ?L - loc)
              (over ?C1 ?C2 - block)
              (at ?I item ?L - loc)
)

(:action pick
      (:parameters ?I - item ?L - location)
      (:preconditions (and (at ?I ?L) (not exists (?I2 item) (picked ?I2))))
      (:effects (and (not at ?I ?L) (picked ?I))
)

(:action pick-block
      (:parameters ?C - block ?C1 - block)
      (:preconditions (and (not exists(?C2 - block) (over ?C2 ?C) (not exists (?I2 item) (picked ?I2)) (over ?C ?C1))
      (:effect (and (not over ?C ?C1) (picked ?C))
)

(:action put
      (:parameters ?I - item ?L - loc)
      (:preconditions (and (picked ?I) (not exists (?I2 - item) (at ?I2 ?L)) (not dirty ?L))
      (:effects (and (not picked ?I) (at ?I ?L))
)

(:action putover
      (:parameters ?C1 ?C2 - block)
      (:preconditions (and (picked ?C1) (not exists (?C3 - block) (over ?C3 ?C2))))
      (:effects (and (over ?C1 ?C2) (not picked ?C1))
)

(:action clean
    (:parameters ?T - tissue ?L - loc)
    (:preconditions (and (picked ?T) (dirty ?L))
    (:effects (not dirty ?L))
)

Problem File

(:Objects     C0 C1 C2 - block
              T - tissue
              P00 P02 P10 P11 P12 - loc
)

(:init  (dirty P10) (at C0 P00) (at C1 P02) (at T P11) (at C2 P12))

(:goal  (at C1 P10) (over C0 C1) (over C2 C0))
```

Now we consider the following plan:

(pick T P11) (clean T P10) (put T P11) (pick C1 P02) (put C1 P10) (pick C0 P00) (putover C0 C1) (pick C2 P12) (putover C2 C0)

To apply local search we must give a state space description. We consider the STRIPS state space where a single state is given by the true predicates, for example:

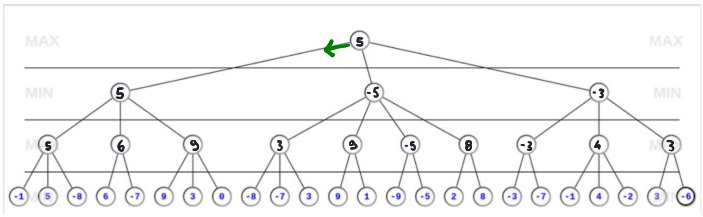


This problem have a small state space, so local search is not the best way to solve the problem. As the evaluation function, we migth use the Following:

S = dirty cell + (3 - number of block on P10) + $\left(\sum_{x \in \text{loc} \setminus \{P10\}} \text{number of block on } x\right)$ + blocks over C2

and apply beam hill climbing by pursuing nodes that have lower value for S.

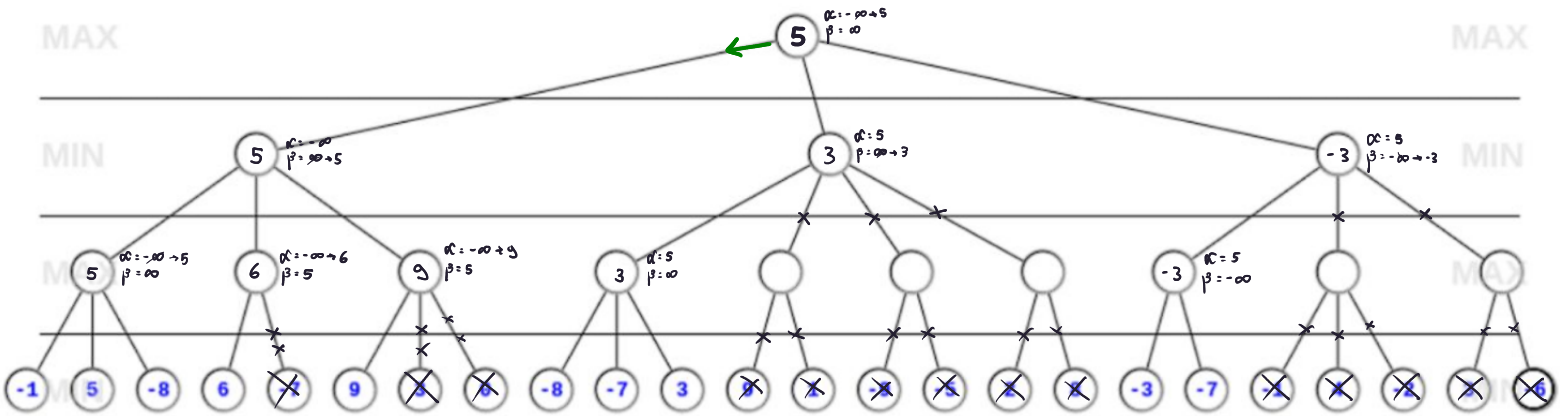
Consider the game tree within the two players, Max and Min, depicted in the figure. Max is the next to move.



a. Run the Minimax algorithm. What is Max's next move? left-branch

b. Run the Alpha-beta pruning algorithm. Show the alpha and beta values at every internal node and how they change throughout the algorithm.

Now with α-β pruning:



1. Erika is the owner of two dogs: Fido and Pluto. Dogs that have owners are happy if their owner feeds them. Erika loves her dogs. Everybody feeds those they love if food is available.
   a. Translate the sentences above in **FOL (First Order Logic)**. Define a vocabulary for the constants, predicates, and functions you need.
   b. Translate the sentences in CNF (Conjunctive Normal Form) showing the transformations of the formulas.
   c. Tell whether it is possible to prove that "Fido and Pluto are happy" **via resolution**. If it is not possible, which knowledge is needed to make such an inference?
2. Given the following two formulas:

$$R(h(X), f(h(b), Y)) \quad \text{and} \quad R(Y, f(Y, h(g(a))))$$

   a. Can they be unified? If so, show the unification process.
   b. If they can be unified, write the Most General Unifier.

   Note: capital letters are variables, and lowercase letters are constants.
3. Given the following KB:

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

   apply the DPPL algorithm with clause learning and show the various iterations of the algorithm. Assume that the variables are selected in alphabetical order and that the splitting rule attempts the value "False" first.

1) $Owner(erika, Fido) \wedge Owner(erika, pluto)$

2) $\forall x \left[[Dog(x) \wedge \exists y (Owner(y, x) \wedge Feeds(y, x))] \rightarrow Happy(x) \right]$

3) $\forall x \left[(Dog(x) \wedge Owner(erika, x)) \rightarrow Loves(erika, x) \right]$

4) $\forall x \forall y \left[(Loves(x, y) \wedge Food) \rightarrow Feeds(y, x) \right]$

Now, transform in CNF

2) $\forall x \left[[Dog(x) \wedge \exists y (Owner(y, x) \wedge Feeds(y, x))] \rightarrow Happy(x) \right]$

$\forall x \exists y \left[[Dog(x) \wedge (Owner(y, x) \wedge Feeds(y, x))] \rightarrow Happy(x) \right]$

$\forall x \left[[Dog(x) \wedge Owner(S(x), x) \wedge Feeds(S(x), x)] \rightarrow Happy(x) \right]$

$\forall x \left[\neg Dog(x) \vee \neg Owner(S(x), x) \vee \neg Feeds(S(x), x) \vee Happy(x) \right]$

3) $\forall x \left[(Dog(x) \wedge Owner(erika, x)) \rightarrow Loves(erika, x) \right]$:

$\forall x \left[\neg Dog(x) \vee \neg Owner(erika, x) \vee Loves(erika, x) \right]$

4) $\forall x \forall y \left[(Loves(x, y) \wedge Food) \rightarrow Feeds(y, x) \right]$

$\forall x \forall y \left[\neg Loves(x, y) \vee \neg Food \vee Feeds(y, x) \right]$

To prove: $Happy(Fido) \wedge Happy(pluto)$ We need to know if Food is available, so we add to the KB: $\{Food\}$. We consider the sentence negated:

$\neg [Happy(Fido) \wedge Happy(pluto)] \Rightarrow \{\neg Happy(Fido), \neg Happy(pluto)\}_5$ the KB is

$\{Owner(erika, Fido)\}_{1A}$ $\{Owner(erika, pluto)\}_{1B}$ $\{Dog(Fido)\}_{1C}$ $\{Dog(pluto)\}_{1C}$

$\{\neg Dog(x_2), \neg Owner(S(x_2), x_2), \neg Feeds(S(x_2), x_2), Happy(x_2)\}_2$

$\{\neg Dog(x_3), \neg Owner(erika, x_3), Loves(erika, x_3)\}_3$

$\{\neg Loves(x_4, y_4), \neg Food, Feeds(y_4, x_4)\}_4$

$\{\neg Happy(Fido), \neg Happy(pluto)\}_5$

$\{Food\}_6$

From 1A and 3 with $S: \left\{\frac{x_3}{Fido}\right\}$: $\{\neg Dog(Fido), Loves(erika, Fido)\}_7$

From 1C and 7: $\{Loves(erika, Fido)\}_8$

From 4 and 8 with $s = \{\frac{x_4}{erika}, \frac{E_4}{Fido}\} : \{\neg Food, Feeds(erika, Fido)\}_9$

From 9 and 6 $\{Feeds(erika, Fido)\}_{10}$ we have: $S(Fido) = erika$ so

From 10 and 2 with $s = \{\frac{x_2}{Fido}\}$:

$\{\neg Dog(Fido), \neg Owner(erika, Fido), Happy(Fido)\}_{11}$

From 11 and 1A and 1C : $\{Happy(Fido)\}_{12}$

From 12 and 5 : $\{\neg Happy(pluto)\}_{13}$

Now we can do the same procedure to derive:

$\{Happy(pluto)\}_{14}$. From 13 and 14 : $\{\square\}_{15} \Rightarrow KB \models (Happy(Fido) \wedge Happy(pluto))$

Now we try to unify $R(h(x), S(h(b), y))$, $R(y, S(y, h(g(a))))$

$D_1 = \{h(x), y\} \Rightarrow s_1 = \{\frac{y}{h(x)}\} \Rightarrow$

$R(h(x), S(h(b), h(x)))$, $R(h(x), S(h(x), h(g(a))))$

$D_2 = \{x, b\}$ $s_2 = s_1\{\frac{x}{b}\} = \{\frac{y}{h(b)}, \frac{x}{b}\}$

$R(h(b), S(h(b), h(b)))$, $R(h(b), S(h(b), h(g(a))))$

$\Rightarrow$ there is no more variable and the two formulas aren't identical, so they can't be unified

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

set $A = F$

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

conflict. We learn $\{A\}$. We set $A = T$

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

we set $B = F$

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

we set $C = 1$ (UP)

$\{$ $\{E\}$ $\{\neg E\}$ ... conflict, we learn $\{B\} \Rightarrow B = T$

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

set $C = T$ (UP)

$\{\{D, E\}, \{\neg D, \neg E\}\} \Rightarrow D = F \Rightarrow \{\{E\}\} \Rightarrow E = T$.

Assignment: $A: T$ $B: T$ $C: T$ $D: F$ $E: T$

Consider the following constraint network: $\gamma = (V, D, C)$.

- Variables: $V = \{a, b, c, d, e, f, g\}$.
- Domains: for all $v \in V$, $D_v = \{1, 2, 3, 4, 5, 6\}$.
- Constraints: $e = 2a$, $b = e$, $c = e + 1$, $d = 2c - 2$, $f = 2e + 1$, $g = f - 2$.

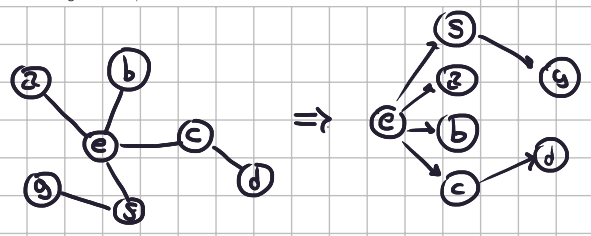
a. Draw the constraint graph γ.

b. Consider running the AcyclicCG algorithm on it. In particular, pick e as the root, draw the directed tree obtained, and give the resulting variable ordering. If the ordering of the variables is not unique, break ties using alphabetical order.

c. Complete the execution of the AcyclicCG algorithm by listing the calls to $\text{Revise}(\gamma, v_{\text{parent}(i)}, v_i)$ and, for each of them, give the resulting domain of $v_{\text{parent}(i)}$.

d. Give the complete assignment of the variables.

e. Is it possible to apply the AC-3 algorithm on γ to find a solution? Give the answer and motivate it.



order For pair wise revision

(c,d) (f,g) (e,f) (e,c) (e,a) (e,b)

Revise c,d ⇒ $D_c = \{2, 3, 4\}$

Revise f,g ⇒ $D_f = \{3, 5\}$

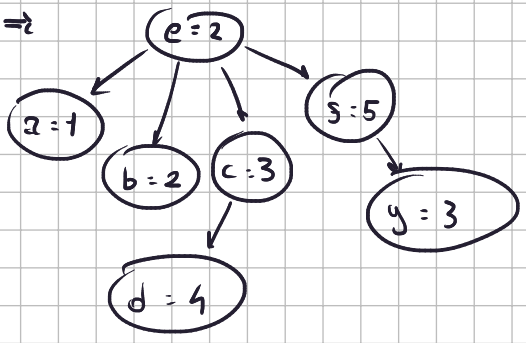
Revise e,f ⇒ $D_e = \{1, 2\}$

Revise e,c ⇒ D_e //

Revise e,a ⇒ $D_e = \{2\}$

Revise e,b ⇒ $D_e = \{2\}$

⇒



Yes, it is possible since the AC3 will make the network arc consistent, by reducing the domain of e to {2}, since the constraint graph is not cyclic, and e is the root node of the spanning tree, the assignment e=2 will implies assignment for all the other variables, so apply AC3 is sufficient to find a solution for the network.