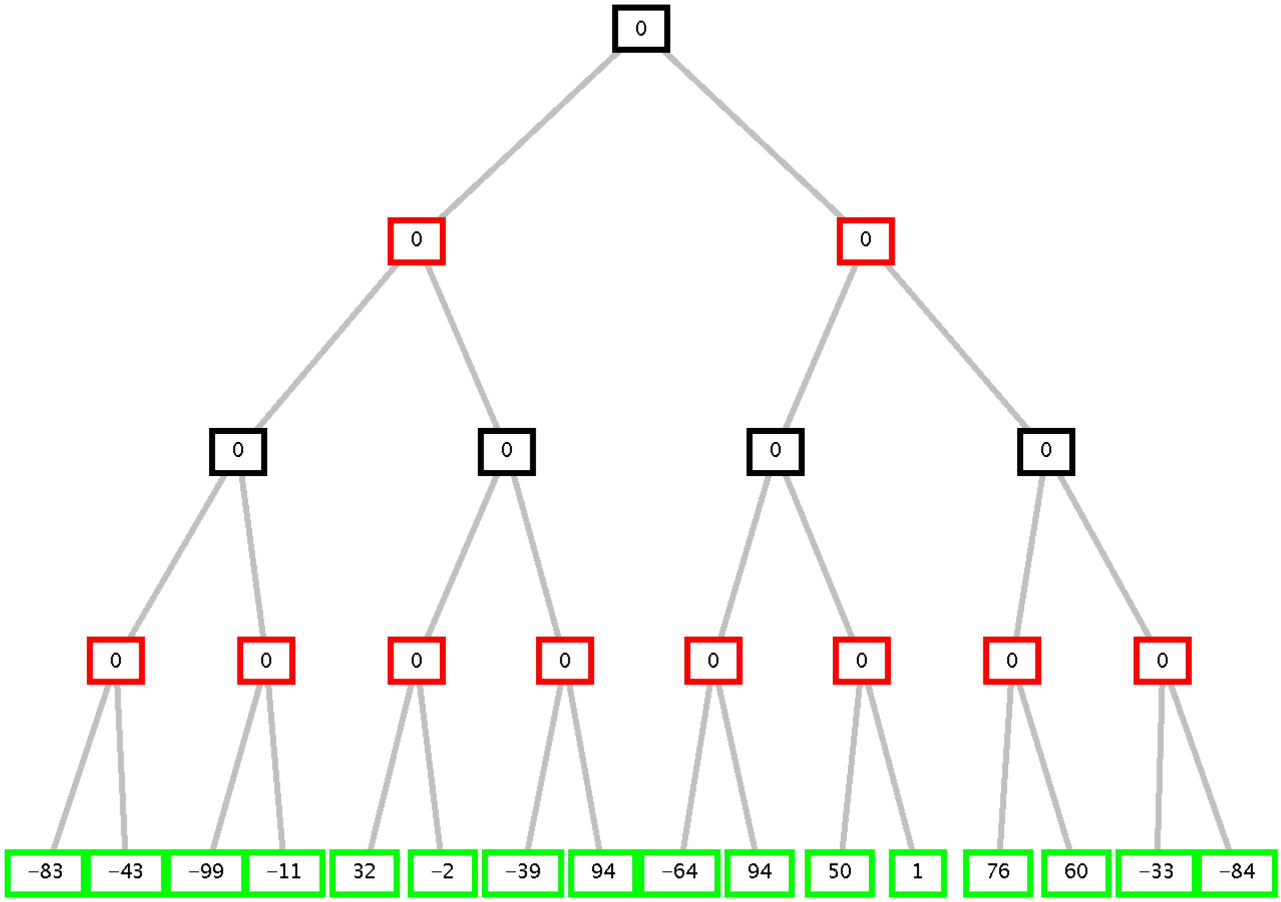
EXAM OF FUNDAMENTALS OF AI – FIRST MODULE

PROF. MICHELA MILANO

Exercise 1

Consider the following game tree where the first player is *MAX*. Show how the *min-max* algorithm works and show the *alfa-beta* cuts. Also, show which is the proposed move for the first player.



Exercise 2

Four friends, Anna, Bea, Carla and Didi, went shopping and buying scratch cards in a mall. We know that in the entrance the friends each had an amount different from those of the others, and equal to 100, or 200, or 300, or 400 euros. Leaving the mall, we take friends to tell each other:

Anna: I earned! I went out with 100 euros more than what Bea had when she entered;

Didi: I don't have a euro anymore! I spent the same sum that Bea spent; but something remained to her, but the one with the most money was among us!

Carla: I offer you something, I earned a sum equal to twice what Didi spent and now I have the same money as Anna has.

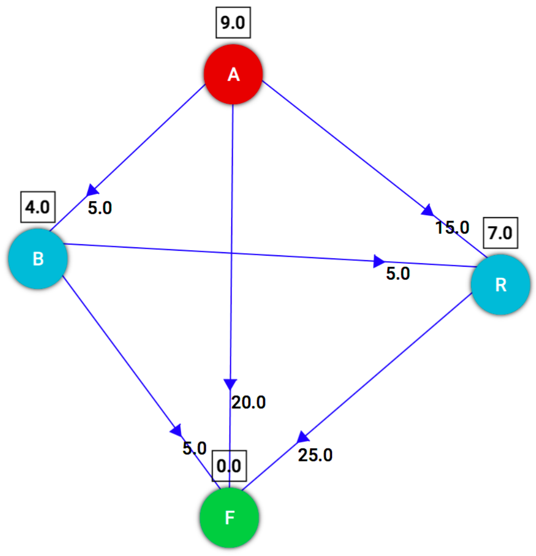
Each of the friends went out with a sum of 0, 100, 200, 300, 400 or 500 euros.

Model the problem as a CSP, where the variables represent the money the friends have before shopping (A, B, C, D, respectively), and the money they have after shopping (A', B', C' and D').

Then apply the arc-consistency technique to the initial network (**consider for this purpose only the unary and binary constraints** in the application of the arc-consistency).

Exercise 3

Consider the following graph, where A is the starting node and F the goal node. The number on each arc is the cost of the operator for the move. Close to each node there is the heuristic evaluation of the node itself, namely its estimated distance from the goal:



1) Show the search tree generated first Best-First and then by the A* algorithm along with the order of expansion of nodes. In case of ties, chose the node to expand in alphabetical order. Consider as the heuristic h(n) the one indicated in the square close to each node in the figure.
2) Is the heuristic admissible?
3) Which is the cost of the path found by Best-First and by A*?

Exercise 4

Consider the problem of moving an object initially on the table to room1 by using a robot.
We have the following actions available:

Loading an object
load (X, Pos)
PREC: at (robot, Pos), at (X, Pos), robotfree
EFFECT: in (robot, X), ¬robotfree

Moving an object
carry (X, Pos1, Pos2)
PREC: at (robot, Pos1), in (robot, X)
EFFECT: at (robot, Pos2), ¬ at (robot, Pos1)

Unloading an object
deliver (X, Pos)
PREC: at (robot, Pos), in (robot, X)
EFFECT: at (X, Pos), ¬ in (robot, X)

Initial state: **at (robot, table), in (robot, ogg)**
Goal state: **at (ogg, room1)**

Show how the STRIPS algorithm finds a solution (only one successful path in the search tree should be shown).

Exercise 5

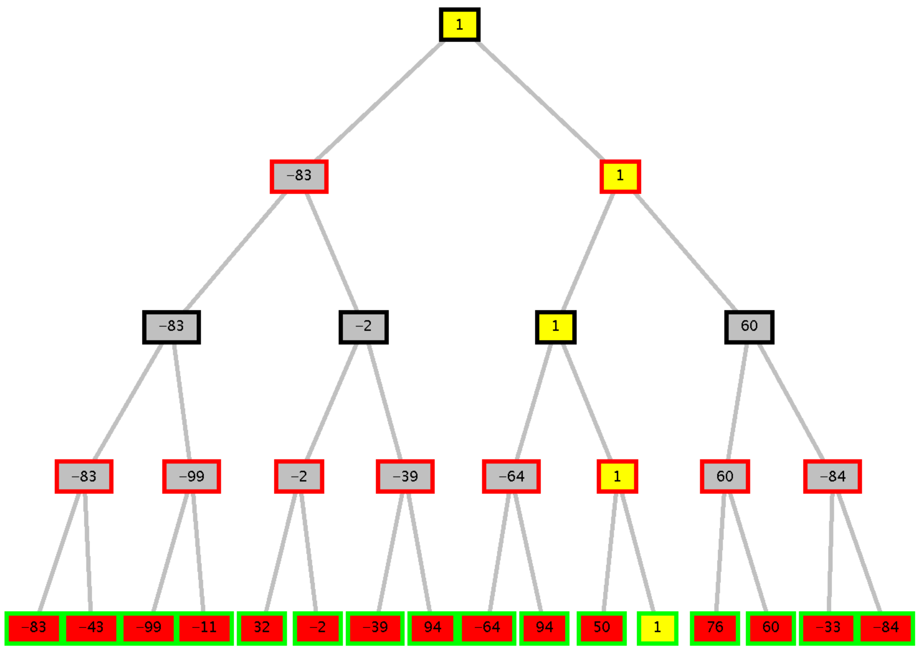
4) Model the action **load** (preconditions, effects and frame axioms), the initial state and the goal of the exercise 4 using the Kowalsky formulation
5) Show two levels of graph plan when applied to exercise 4.
6) What is hierarchical planning and explain the method presented during the course.
7) What is Particle Swarm Optimization and which are its main features?

8) What are non-informed search strategies? Describe the strategies that have been presented during the course.

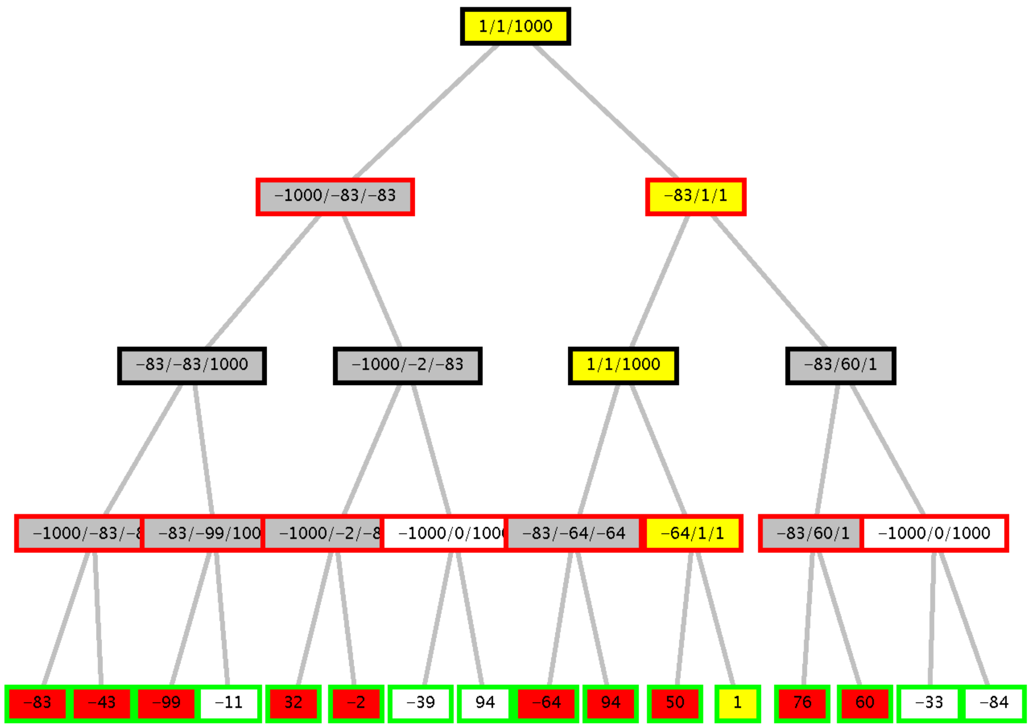
Solution

Exercise 1

Min-max



Alfa-beta:



Exercise 2

Variables: initial amount of money for each friend A, B, C, D in input and final amount of money A’, B’, C’, D’

Domains: (1..4) for A, B, C, D and integer value for A’, B’, C’, D’
A:: [1,2,3,4]
B:: [1,2,3,4]
C:: [1,2,3,4]
D:: [1,2,3,4]
A’:: [0,1,2,3,4,5]
B’:: [0,1,2,3,4,5]
C’:: [0,1,2,3,4,5]
D’:: [0,1,2,3,4,5]

Constraints:
Variables A, B, C, D differents:
$A \neq B \neq C \neq D$

Anna (and Bea):
$A' > A$
$A' = 1 + B$

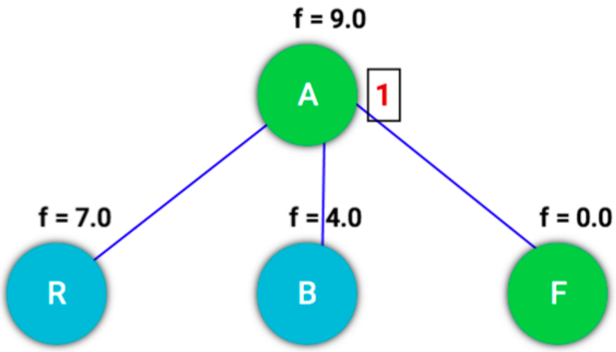
Didi:
$D' = 0$
$D - D' = B - B'$
$D' < D$
$B' < B$
B = 4 (>A,C,D)
$B' > 0$

Carla:
$C' > C$
$C' - C = 2^* (D - D') = 2^*D$ *essendo* $D'=0$
$C' = A'$

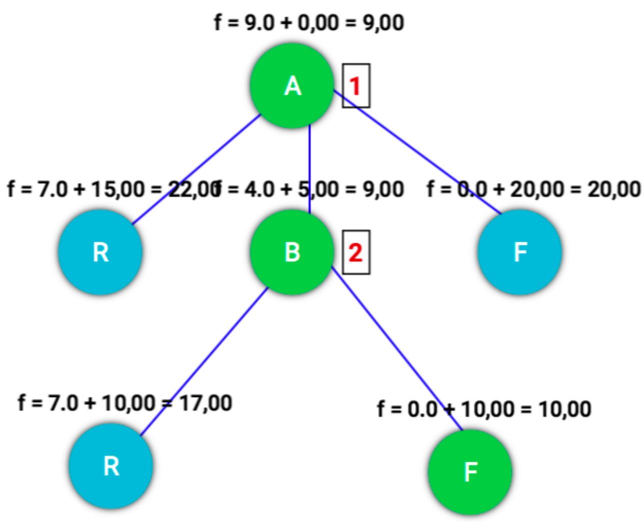
Arc-consistency
A:: [1,2,3]
B:: [4]
C:: [1,2,3]
D:: [1,2,3]
A’:: [5]
B’:: [1, 2, 3]
C’:: [5]
D’:: [0]

Exercise 3

Best-first (cost of the path: 20)



A* (Admissible heuristics; cost of the path: 10)



Exercise 4

STATO	GOAL
at (robot,table) in(robot,ogg)	at(ogg,room1)

L'unica azione che ha un effetto che puo' unificare con at(ogg,room1) e' la deliver

STATO	GOAL
at (robot,table) in(robot,ogg)	at(robot,room1) in (robot, ogg) at(robot,room1) ∧ in (robot, ogg) **deliver(ogg,room1)**

at (robot, room1) posso raggiungerla con carry

STATO	GOAL
at (robot,table)	in(robot,X)
in(robot,ogg)	at(robot,Pos1)
	in(robot,X) ∧ at(robot,Pos1)
	carry(X,Pos1,room1)
	in (robot, ogg)
	at(robot,room1) ∧ in (robot, ogg)
	deliver(ogg,room1)

Entrambe le precond di carry sono verificate legando X/ogg e Pos1 con table

STATO	GOAL
at (robot,table)	**carry(ogg,table,room1)**
in(robot,ogg)	in (robot, ogg)
	at(robot,room1) ∧ in (robot, ogg)
	deliver(ogg,room1)

Le precondizioni di deliver sono soddisfatte nello stato

STATO	GOAL
at (robot,room1)	**deliver(ogg,room1)**
in(robot,ogg)	

STATO	GOAL
at (robot,room1)	
at(ogg,room1)	

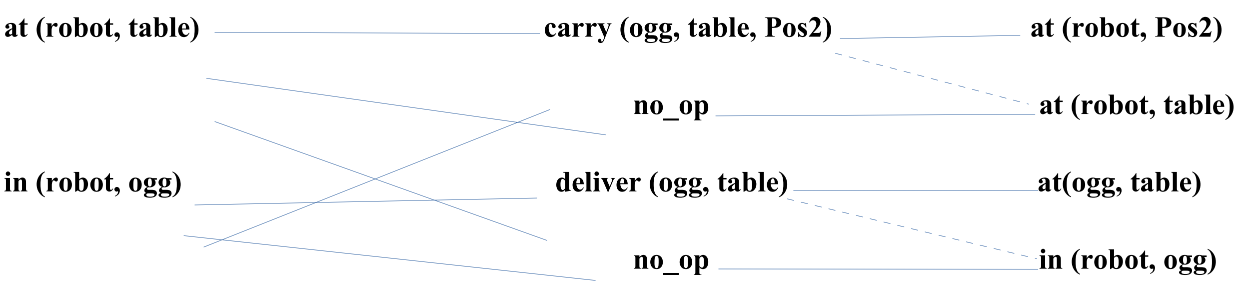
Exercise 5

1)

holds(at(robot,table),s0).
holds(in(robot,ogg), s0).

holds(in(robot,X), do(load(X,Pos),S))
pact(load(X,Pos),S):- holds(at(robot,Pos), S), holds(at(X,Pos),S), holds(robotfree,S).
holds(V,do(load(X,Pos),S)):- holds(V,S), V\=robotfree.

2)


at (robot, table)
in (robot, ogg)
carry (ogg, table, Pos2)
no_op
deliver (ogg, table)
no_op
at (robot, Pos2)
at (robot, table)
at(ogg, table)
in (robot, ogg)