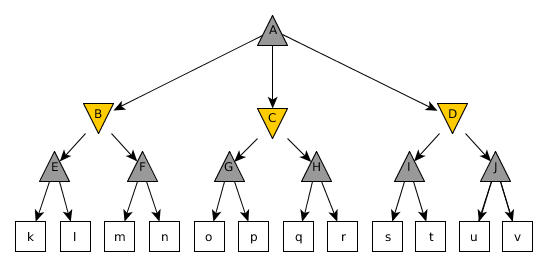
Introduction into Artificial Intelligence

Mid-term 2. – Group A

1. Solve these problems on paper:



1. The evaluation function assigns the following numbers to the leaves of the tree: [[[6, 0], [-2, 1]], [[-3, -5], [3, 5]], [[-7, -4], [4, -6]]]. Give a good strategy for the ‘Maximum’ player using the Mini-max algorithm.
2. The evaluation function assigns the following numbers to the leaves of the tree: [[[-7, -3], [1, -2]], [[4, -1], [-6, 5]], [[-5, -4], [6, 2]]]. Which nodes will not have a value determined because of pruning? (The nodes in the figure are examined from left to right.)
3. We have constraints A, B, C, D, E, F ∈ {1,2,3,4,5,6}, 3 \* A – 1 < C, A < B, B = 3 \* E, 3 \* E > D, F < D, C < F. Give the possible domain for each variable after checking arc consistency.
4. There are 26 slots in a row on the table. 2 players take turns on placing down 1 rock in one of the slots. A player can only place a rock in a blank slot where its left neighbor is also blank. The player who would be next and can’t put a rock in a slot loses the game. Implement this game using the provided name\_neptuncode.py file. Follow the instructions in the code. (5 points)
5. **Description:**

H= {0, 1}

0 – blank slot, 1 – there is a rock in the slot

1. **Set of game states:**

*(if there is a rock in a slot, the left neighbor should be blank)*

1. **Initial game state:**

1. **End game state:**

*(for each blank slot there is a rock in the left neighbor, meaning the player can’t place the next rock)*

1. **Set of steps:**

*(the slot where we want to put the rock is empty and left neighbor slot is also empty)*

1. **Effect definition:**