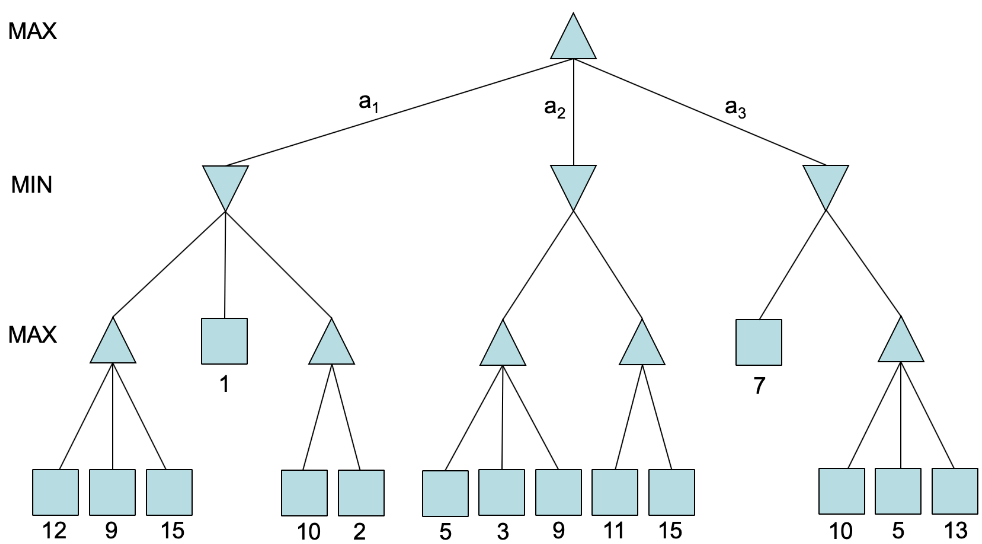
Washington State University
School of Electrical Engineering and Computer Science
Fall 2019

CptS 440/540 Artificial Intelligence
Homework 4 – Solution
Due: September 19, 2019 (11:59pm)

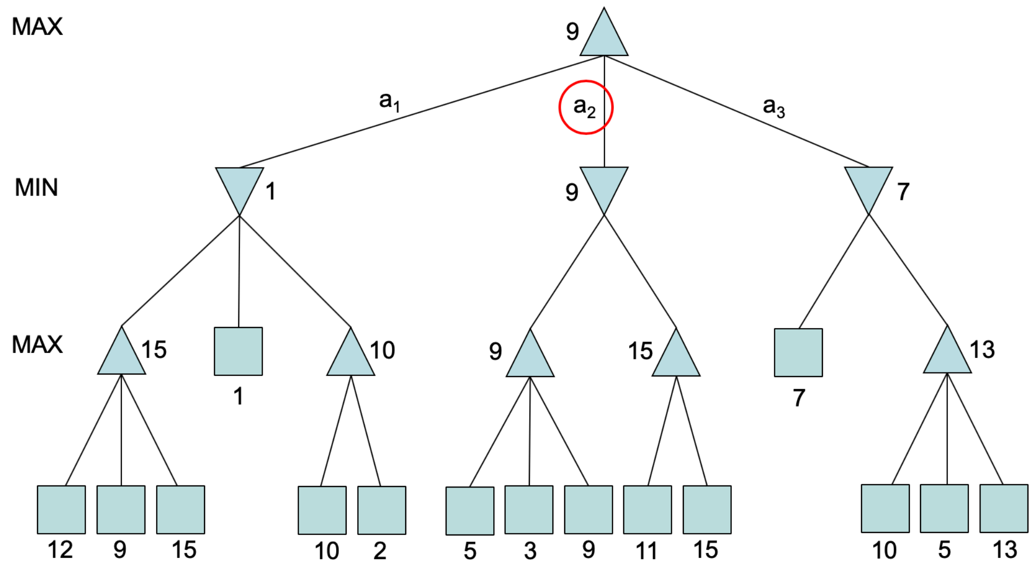
General Instructions: Put your answers to the following problems into a PDF document and submit as an attachment under Content → Homework 4 for the course CptS 440 Pullman (all sections of CptS 440 and 540 are merged under the CptS 440 Pullman section) on the Blackboard Learn system by the above deadline. Note that you may submit multiple times, but we will only grade the most recent entry submitted before the above deadline.

1. Consider the following game tree. Upward-pointing triangles are MAX nodes, downward-pointing triangles are MIN nodes, and squares are terminal nodes.



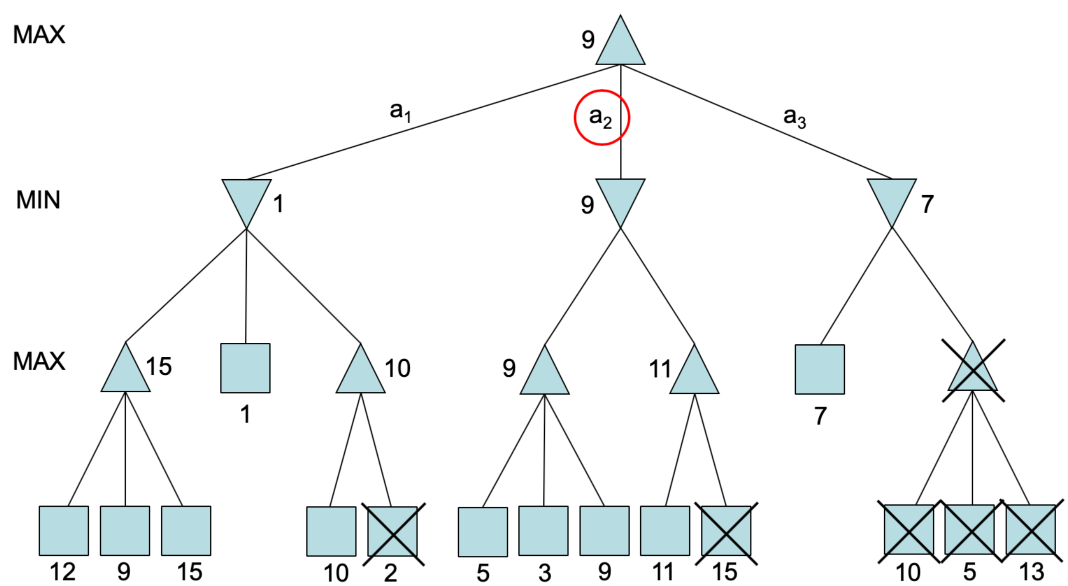
a. Perform Minimax-Decision search on the above tree. Put the final value next to each node in the tree. Finally, indicate which action MAX should take: a_1, a_2 or a_3.

Solution:



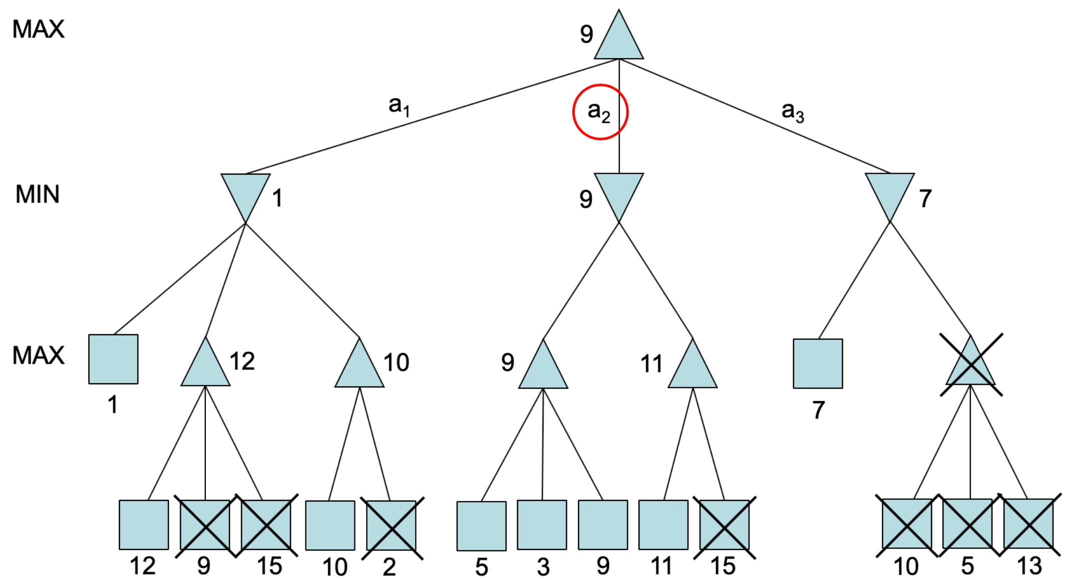
b. Perform Alpha-Beta-Search on the above tree (don't reuse your tree from part (a)). Put an "X" over all nodes (internal and terminal, and all nodes in a subtree) that are pruned, i.e., not evaluated. Put the final value next to all non-pruned nodes. Finally, indicate which action MAX should take: a_1, a_2 or a_3.

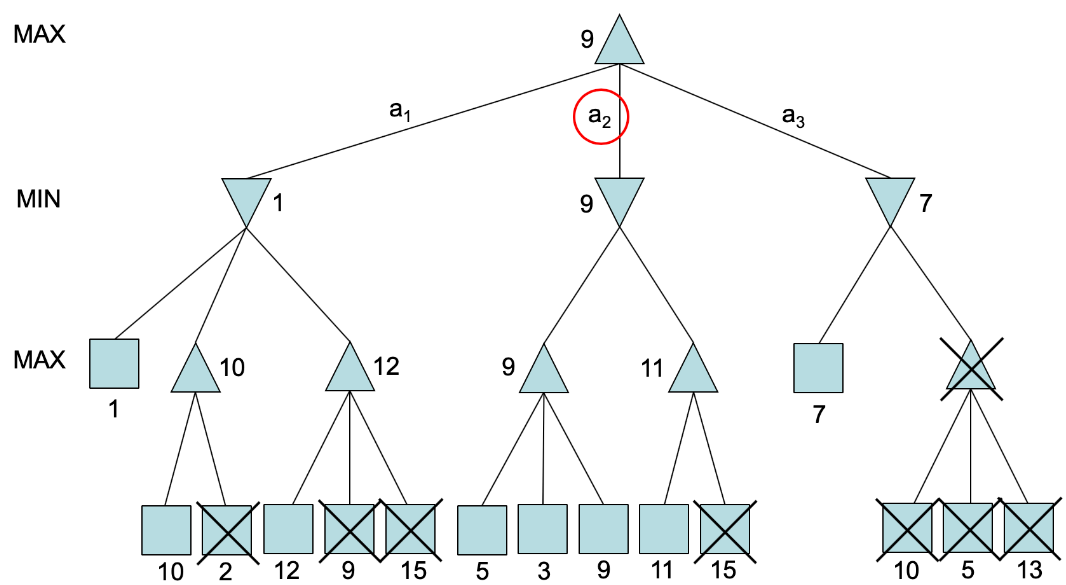
Solution:



c. *540 students only*. Consider reordering the children of the MIN nodes in the above tree in order to maximize the number of nodes pruned by the Alpha-Beta-Search. You cannot reorder the children of any other nodes above or below the MIN nodes. Show this tree and perform Alpha-Beta-Search on this tree in the same way as described in part (b).

Solution: The subtrees on the middle and right MIN nodes should remain the same; shuffling them would decrease pruned nodes. However, for the left MIN node, moving the terminal node (utility=1) to the left allows pruning the 9 and 15 terminal nodes under that MAX node. The non-terminal nodes under the left MIN node can be in either order. So, the following two trees both lead to the maximum amount of pruning.





2. Consider the following logic problem:

   If you like checkers, then you like chess.
   If like computers, then you like coding.
   If you like chess and you like coding, then you will learn AI.
   If you learn AI, you will be rich and famous.
   You like checkers.
   You like computers.
   Prove that you will be rich.

   a. We will solve this problem using propositional logic. First, show one propositional logic sentence for each of the first six sentences in the above problem. You may only draw from the following atomic sentences.
      - Like(Checkers)
      - Like(Chess)
      - Like(Computers)
      - Like(Coding)
      - Learn(AI)
      - Rich
      - Famous

      Solution:
      i. Like(Checkers) $\Rightarrow$ Like(Chess)
      ii. Like(Computers) $\Rightarrow$ Like(Coding)
      iii. Like(Chess) $\wedge$ Like(Coding) $\Rightarrow$ Learn(AI)
      iv. Learn(AI) $\Rightarrow$ Rich $\wedge$ Famous
      v. Like(Checkers)
      vi. Like(Computers)

   b. Convert each of the six sentences from part (a) into Conjunctive Normal Form (CNF). You may just show the final result for each sentence; no need to show the intermediate steps. Number your clauses.

      Solution:
      CNF
      i. $\neg$Like(Checkers) $\vee$ Like(Chess)
      ii. $\neg$Like(Computers) $\vee$ Like(Coding)
      iii. $\neg$Like(Chess) $\vee$ $\neg$Like(Coding) $\vee$ Learn(AI)
      iv. ($\neg$Learn(AI) $\vee$ Rich) $\wedge$ ($\neg$Learn(AI) $\vee$ Famous)
      v. Like(Checkers)
      vi. Like(Computers)

Clauses:

1. ¬Like(Checkers) ∨ Like(Chess)
2. ¬Like(Computers) ∨ Like(Coding)
3. ¬Like(Chess) ∨ ¬Like(Coding) ∨ Learn(AI)
4. ¬Learn(AI) ∨ Rich
5. ¬Learn(AI) ∨ Famous
6. Like(Checkers)
7. Like(Computers)

c. Perform a resolution proof by refutation to prove you will be Rich, using the knowledge base from part (b). For each resolution step, show the numbers of the clauses used, the resulting clause, and then number the resulting clause.

Solution: Add to the KB the negation of the sentence to be proved as clause 8:

8. ¬Rich

Resolve clauses 8 and 4:

9. ¬Learn(AI)

Resolve clauses 9 and 3:

10. ¬Like(Chess) ∨ ¬Like(Coding)

Resolve clauses 10 and 2:

11. ¬Like(Chess) ∨ ¬Like(Computers)

Resolve clauses 11 and 1:

12. ¬Like(Checkers) ∨ ¬Like(Computers)

Resolve clauses 12 and 6:

13. ¬Like(Computers)

Resolve clauses 13 and 7:

Empty clause

Thus, the negated query (¬Rich) must be false. Therefore, the query (Rich) must be true.

d. *540 students only*: What one literal (other than ¬Famous) could you add to the knowledge base in part (b) in order to prove ¬Famous using resolution proof by refutation?

Solution: Any literal that is the negation of a literal already in the KB, or entailed by the KB, can be added to the KB in order to make it inconsistent, and therefore able to prove anything (namely, ¬Famous) by using the inconsistency to derive the empty clause. For example, adding the literal ¬Like(Computers) to the KB would introduce such an inconsistency. Other possibilities are ¬Rich, ¬Like(Checkers), ¬Like(Chess), ¬Like(Coding), ¬Learn(AI).