

Homework #3: Game Trees & Logic

85 Points Total (53 in L^AT_EX and 32 in .py)

Instructor: Haym Hirsh

Name: Student name, Netid: NetId

Course Policy: Read all the instructions below carefully before you start working on the assignment, and before you make a submission. **These can be specific to each assignment.**

- Please include your name and NetIDs on the first page. We recommend typesetting your submission in L^AT_EX, and an Overleaf template is linked [here](#). See course policy for general typesetting requirements.
- Homeworks must be submitted via Gradescope by the due date and time. This assignment requires **two individual submissions**: one to “Homework 3 - Written” and one to “Homework 3 - Python.” Your submission will be marked on time if and only if both assignments are submitted by the due date and time.
- Late homeworks are accepted until **3/14/20 at 11:59pm** for a 50% penalty per course policy.
- All sources of material outside the course must be cited. The University Academic Code of Conduct will be strictly enforced.

Problem 1: Unification and Resolution

(20 points)

A (6 points): A key component to first-order inference algorithms is the UNIFY algorithm. Recall that this algorithm returns a substitution that would make two logical expressions look identical. If such a substitution doesn't exist then return *failure*. Do the following unifications and give the most general unifier as your answers:

1. $\text{UNIFY}(\text{loves}(\text{Haym}, x), \text{loves}(y, \text{AI})) =$

[Your Solution Here]

2. $\text{UNIFY}(\text{loves}(\text{Haym}, x), \text{loves}(x, \text{AI})) =$

[Your Solution Here]

3. $\text{UNIFY}(\text{loves}(\text{AI}, x_1), \text{loves}(x_2, x_3)) =$

[Your Solution Here]

B (4 points): Convert the following statement into conjunctive normal form (CNF).

$$\neg P_1 \vee (P_2 \Rightarrow (\neg P_3 \wedge P_4))$$

[Your Solution Here]

C (10 points): Given the following knowledge base,

$$\forall x [(A(x) \wedge B(x)) \Rightarrow C(x)] \quad \forall x \forall y [C(x) \Rightarrow (D(y) \vee E(y))],$$

do a proof by resolution to prove

$$\forall x \forall y [(A(x) \wedge B(x)) \Rightarrow (D(y) \vee E(y))].$$

Please show *all* steps (e.g., negation of the conclusion, conversion to CNF, which clauses are being resolved at each step and the *unifiers* that were used to do so, etc.).

[\[Your Solution Here\]](#)

Problem 2: Minimax alpha-beta pruning

(10 points)

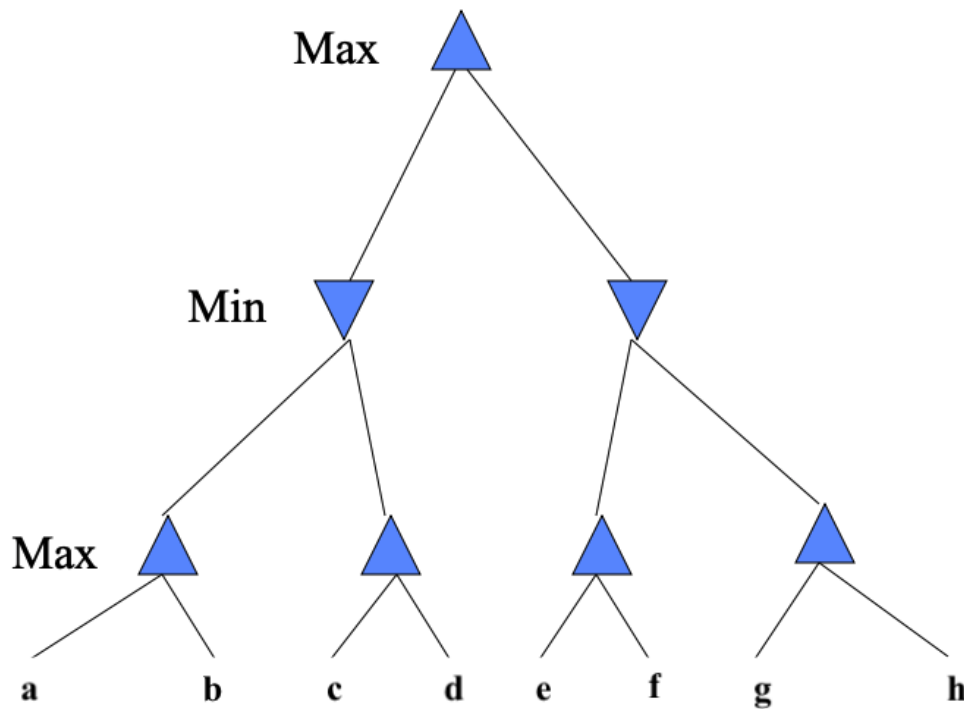


Figure 1: Min-max tree for Problem 2

Consider the above min-max tree in Figure 1, **a** to **h** represent numbers on the leaves. You should traverse the left sub-tree before the right sub-tree.

A (5 points): What criteria shall **a** to **h** hold if no pruning is possible? (Hint: the solution should be a set of inequalities.)

[\[Your Solution Here\]](#)

B (5 points): What is the maximum number of leaves that can be pruned? What criteria should **a** to **h** hold if maximum pruning occurs? (Hint: the solution should again be a set of inequalities.)

[\[Your Solution Here\]](#)

Problem 3: Minimax alpha-beta pruning, again

(15 points)

Consider the min-max tree in Figure 2. The values for each leaf are shown in the last row.

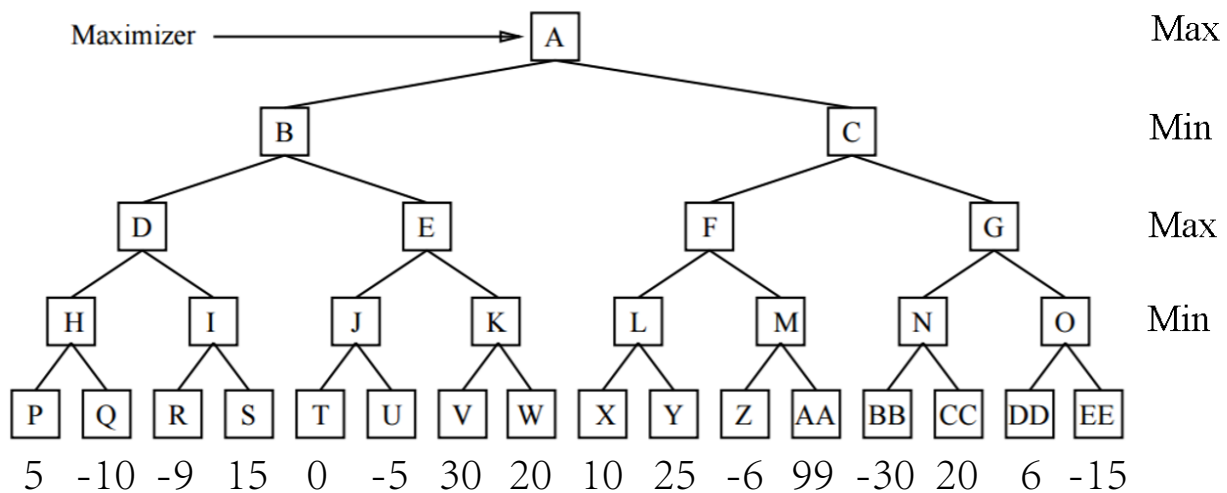


Figure 2: Min-max tree for Problem 3

A (5 points): Which move should be made at the top level (Left or Right)? What is its value? Show the steps of applying the Minimax algorithm below by giving the specific value for each node. (For example, $P = 3, Q = -10$...)

[Your Solution Here]

For subproblems B & C, report your answers by writing:

1. the edge(s) that would be cut (e.g. M-to-AA)
2. the values of alpha and beta for each node that gets pruned at the time of the pruning.

B (5 points): Which nodes get pruned if you use alpha-beta pruning when you explore successors of states going left-to-right?

[Your Solution Here]

C (5 points): Which nodes get pruned if you use alpha-beta pruning when you explore successors of states going right-to-left?

[Your Solution Here]

Problem 4: SAT Solvers

(40 points)

The information for this problem can be found in a Jupyter Notebook. You can access it by using one of the following resources:

- Go to this (read only) Google Colab link: <https://colab.research.google.com/drive/1pmGpgJf6fAm5mUkrGwNrpqR9dYhMfsHJ>. Then press “Open in Playground” and then “Copy to Drive” (both in the top left corner) to make your own copy.
- Download the .ipynb file from the course website here: <https://www.cs.cornell.edu/courses/cs4700/2020sp/homeworks/hw3-SAT.ipynb>

A (8 points): See .ipynb file for question statement.

[Your Solution Here]

B (18 points): Provide solution in .py file; submit via second Gradescope assignment.

C (14 points): Provide solution in .py file; submit via second Gradescope assignment.