University of Waterloo
School of Computer Science
CS 486/686, SAMPLE Midterm Examination
Winter 2025

Name:

Waterloo Student ID:

- **Instructor: Victor Zhong & Jesse Hoey**
- **Date: Any day before Feb 13th 700pm**
- **Location: Anywhere you want**
- **Time: Anytime before Feb 13th 700pm**
- **There are 4 questions on this exam**
- **There are 10 pages in this exam (including cover page and two additional pages at the end for work that does not fit in the spaces provided)**
- **There are a total of 100 marks on the exam**
- **You have 110 minutes to complete the exam**
- **ONLY non-programmable calculators are allowed**

Question	1	2	3	4	TOTAL
Marks	20	30	30	20	100
Score					

Question 1: Short Answers (20 marks out of a total of 100 marks)

WRITE ANSWERS IN THE SPACES PROVIDED AFTER EACH QUESTION

(1a) [**4 marks**] Complete the following sentence: A* search uses a priority queue of nodes ranked by _____, and so is a mixture of _____ and best-first search.

(1b) [**4 marks**] A heuristic for a search problem is an estimate of the true cost to the goal. Is the following statement true or false: *an admissible heuristic is always greater than the true cost.* Briefly explain

(1c) [**4 marks**] True or False: in a deterministic planner, causal rules specify when a feature keeps its value when not acted upon. Briefly explain why.

(1d) [**4 marks**] Why is variable elimination for constraint satisfaction problems intractable?

(1e) [**4 marks**] Give one reason why Heuristic Depth-First Search is often used in practice.

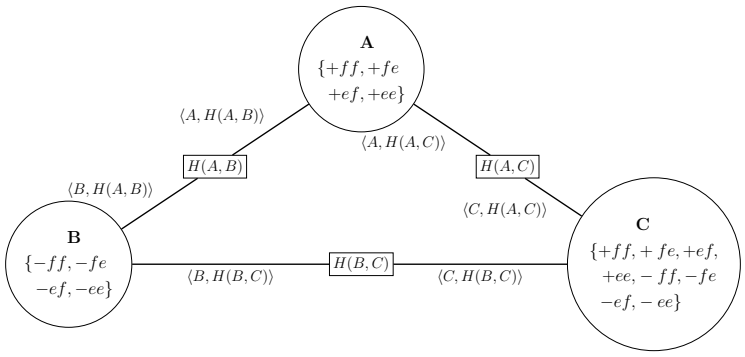
Question 2: Constraint Satisfaction (30 marks marks out of a total of 100 marks)

The KoobeCaf™ social network consists of a number of *members* who can be connected (or not) to each other. Connected members on KoobeCaf™ can be either both *friends* (f) or *enemies* (e) with each other. Members come in two types: positive ($+$) and negative ($-$). While connected members of different types *must* be enemies, members of the same type can be friends or enemies. A *stable* social network is one in which all members satisfy these constraints. This can be represented as a constraint satisfaction problem (CSP) as follows:

• **variables and domains**: Each KoobeCaf™ member is represented with a tuple containing: their type ($+$ or $-$) and whether they are friends (f) or enemies (e) with each of their (up to $N-1$) connections. Thus, a member b of a network with three fully connected members a, b, c (in that order), would be represented with a variable $B = T_b X_a X_c$ where $T_b \in \{+, -\}$ is b's type and $X_a, X_c \in \{f, e\}$ are b's relationship with a and c, respectively. Thus, B has domain $\{+ff, +fe, +ef, +ee, -ff, -fe, -ef, -ee\}$, where $-fe$ means b is type negative, friends with a, but enemies with c. The order of the friend/enemy relations in each tuple is alphabetic.

• **constraints**: A binary constraint between each pair of connected members that requires the two members to (1) have the same relation (e.g. in the 3-member network A,B,C above, if $A = **f$ then $C = *f*$ where $*$ means either value); and (2) be of the same type OR be enemies. Call this constraint $H(\cdot, \cdot)$. For example, with three fully connected members represented by variables A, B, C: $H(A, C) = True$ for
$\{A, C\} \in \{\{+*f, +f*\}, \{+*e, +e*\}, \{-*f, -f*\}, \{-*e, -e*\}, \{+*e, -e*\}, \{-*e, +e*\}\}$.

This graph shows a KoobeCaf™ network with three members: a, b and c, in which a is positive ($+$), and b is negative ($-$) and domain values of the wrong type have been removed from the domains of A and B.



(2a) [**25 marks**] Using AC-3, make the CSP shown above arc-consistent by filling in the table on the facing page, in which each iteration is on a single line in the table and a check-mark is under each constraint that is consistent (i.e. the arc is **not** in the

(Question 3 CONTINUES ON NEXT PAGE...)

TDA queue), and the domain values remaining for each variable are shown under each variable name. Always choose the left-most (in the table) inconsistent constraint to make consistent next. You may not need all rows, but you will not need more rows. Don't add redundant arcs back on the TDA (after changing the domain of X for the arc $\langle X, c(X,Y)\rangle$, you only add arcs $\langle Z, c'(Z,X)\rangle$ to TDA where $\mathbf{Z} \neq \mathbf{Y}$).

$\mathbf{A} = T_a X_b X_c$ $T_a = +$ $X_b, X_c \in \{f, e\}$	$\langle A, H(A,B)\rangle$	$\langle B, H(A,B)\rangle$	$\mathbf{B} = T_b X_a X_c$ $T_b = -$ $X_a, X_c \in \{f, e\}$	$\langle B, H(B,C)\rangle$	$\langle C, H(B,C)\rangle$	$\mathbf{C} = T_c X_a X_b$ $T_c \in \{+, -\}$ $X_a, X_b \in \{f, e\}$	$\langle C, H(A,C)\rangle$	$\langle A, H(A,C)\rangle$
$+ff$ $+fe$ $+ef$ $+ee$			$-ff$ $-fe$ $-ef$ $-ee$			$+ff$ $+fe$ $+ef$ $+ee$ $-ff$ $-fe$ $-ef$ $-ee$		
$+ef$ $+ee$	✓							

(2b) [**5 marks**] Based only on your arc-consistent network (last line of the table), can you guarantee that there is a solution (a stable network)? Briefly explain why or why not.

Question 3: Search (30 marks marks out of a total of 100 marks)

WRITE ANSWERS IN THE SPACES PROVIDED AFTER EACH QUESTION

An artificial intelligence conference has $N_c + N_s$ attendees, of which N_c do research on connectionist AI ("connectionists") and N_s do research on symbolic AI "symbolicists". On their way to the conference reception, they come across a river with no bridge that they must cross. There is a boat that can carry only 2 people at a time. If more connectionists than symbolicists are left on either side of the river, the connectionists will start a fight with the symbolicists. At least one person must be in the boat on each trip (to paddle the boat). The goal is to move everyone to the other side of the river in the shortest number of trips without a fight starting. Assume $N_c < N_s$ to start with.

The state space can be easily described by the number of connectionists and symbolicists on each side, plus the location of the boat. The cost function, $g(n)$ is simply 1 for each trip across the river. A simple heuristic $h(n)$ for this problem is to relax it by ignoring fights that break out.

(3a) [**10 marks**] Give a formal (mathematical) definition of a state, neighborhood function, cost function and the heuristic function described above.

(3b) [**5 marks**] Is the $h(n)$ defined above admissible? Carefully explain why or why not.

(4c) [**15 marks**] Show part of the search graph that results from applying algorithm A* for the problem with $N_c = N_s = 3$ starting from the configuration where everyone is on one side, with the heuristic $h(n)$ and cost $g(n)$ defined above. Specifically, continue until you have expanded (generated the successors of) four nodes. Break ties arbitrarily. Number the four nodes you expand to indicate the order in which they are expanded ("1","2","3","4"). Then, number (with "5") the next leaf that you would expand if you were to continue with A* search on this graph. Label each node with its $g(n)$ value and its $h(n)$ value. You should label all nodes, not just the ones you expand. Do not add a node to the tree if it is already in the tree somewhere with a lower or equal value of $f(n) = g(n) + h(n)$. Branches of your search graph should terminate (have an infinite heuristic value) if the connectionists outnumber the symbolicists on either side of the river.

Question 4: Decision Trees (20 marks out of a total of 100 marks)

You are trying to build a system to predict whether a stock price will rise or fall, using the following dataset of historical measurements of four attributes: the `season`, whether an `election` is taking place, whether the price of oil (`oil_price`) has risen or fallen recently whether the price of the stock (`stock_price`) in question rose or fell.

election	season	oil_price	stock_price
no	winter	rise	rise
no	summer	fall	fall
no	summer	rise	rise
yes	winter	rise	fall
no	winter	fall	fall
yes	summer	rise	rise
yes	summer	fall	fall
yes	winter	fall	fall

(4a) [**15 marks**] Using information gain (weighted by fractions of documents) as your feature selection mechanism, construct a decision tree that predicts if the stock price will rise or fall based on the attributes above. Continue splitting until no more attributes are available, or until all data agrees on the target attribute (stock price change). Show your work and draw a graph of your final decision tree.

(4b) [**3 marks**] What prediction would your decision tree give for the following test data

election	season	oil_price	stock_price
no	summer	fall	
no	winter	rise	
yes	summer	rise	

(4c) [**2 marks**] Suppose now you received the following data points and added them to the data in Q(1), and re-learned the decision tree using information gain to split.

election	season	oil_price	stock_price
yes	summer	rise	fall
yes	winter	rise	rise

Would your decision tree change? If so, what would the change be? Would your classification of the data in (b) change? If so, show the new classification. If not, explain why not.

ADDITIONAL WORK - Clearly state which question you are working on

ADDITIONAL WORK - Clearly state which question you are working on