University of Waterloo
School of Computer Science
CS 486/686, SAMPLE Midterm Examination
Spring 2022

Name:

Waterloo Student ID:

- **Instructor: Jesse Hoey**
- **Date: Any day before June 8th 7pm**
- **Location: Anywhere you want**
- **Time: Anytime before June 8th, 7pm**
- **There are 4 questions on this exam**
- **There are 11 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**	**20**	**30**	**30**	**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.

> **SOLUTION**:
> cost to the node + heuristic from node to goal, lowest-cost-first

(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

> **SOLUTION**:
> false. It must be less than or equal to the true cost

(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.

> **SOLUTION**:
> False. Causal rules specify when things change as a result of an action.

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

> **SOLUTION**:
> It is hard to find the optimal elimination ordering.

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

> **SOLUTION**:
> Space complexity

Question 2: Propositional Logic (20 marks out of a total of 100 marks)

WRITE ANSWERS IN THE SPACES PROVIDED AFTER EACH QUESTION

Consider the following argument

Pablo will go to jail if he breaks the law
If Pablo breaks the law, then he'll get rich
If Pablo is rich, then he won't go to jail
Therefore, Pablo doesn't break the law

(2a) (**3 points**) Assign a set of propositional variables to the statements in this argument.

SOLUTION:
L: Pablo breaks the law
J: Pablo goes to jail
R: Pablo is rich

(2b) (**3 points**) Write the premises as logical statements using your variables from part (a)

SOLUTION:
$L \to J$
$L \to R$
$R \to \neg J$

(2c) (**1 point**) write the conclusion as a logical statement using your atoms

SOLUTION:
$\neg L$

(2d) (**4 points**) Write a conjunction of the premisses and the refutation of the conclusion in Conjunctive Normal Form (CNF)

SOLUTION:

$$\{\{\neg L, J\}, \{\neg L, R\}, \{\neg R, \neg J\}, \{L\}\}$$

(2e) (**4 points**) Prove the validity of the argument using resolution on your CNF

SOLUTION:
resolve second with third to get

$$\{\{\neg L, J\}, \{\neg L, \neg J\}, \{L\}\}$$

resolve the first and second

$$\{\{\neg L\}, \{L\}\}$$

this is a contradiction

(2f) (**5 points**) Prove the validity of this argument using a truth table

SOLUTION:
`all P` means `P1`∧`P2`∧`P3`

L	J	R	P1	P2	P3	all P	D
*F	F	F	T	T	T	T	T
*F	F	T	T	T	T	T	T
*F	T	F	T	T	T	T	T
F	T	T	T	T	F	F	T
T	F	F	F	F	T	F	F
T	F	T	F	T	T	F	F
T	T	F	T	F	T	F	F
T	T	T	T	T	F	F	F

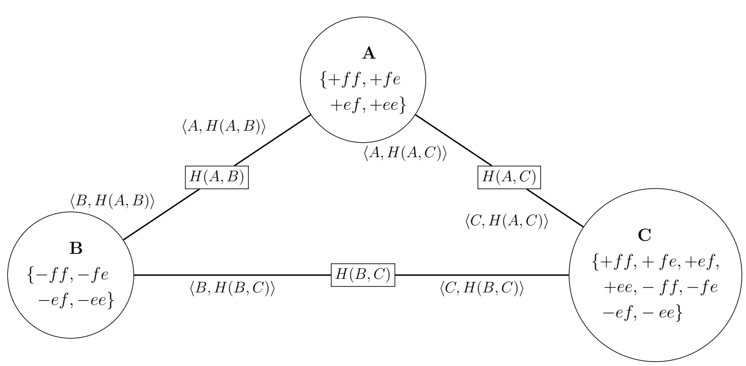
Starred rows have premises true, and every starred row has D true

Question 3: 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.



(3a) [**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 $\mathbf{Z} \neq \mathbf{Y}$).

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

SOLUTION:

10 things change (shown in green) - 2.5 marks per item changed

A	$\langle A, H(A,B)\rangle$	$\langle B, H(A,B)\rangle$	B	$\langle B, H(B,C)\rangle$	$\langle C, H(B,C)\rangle$	C	$\langle C, H(A,C)\rangle$	$\langle A, H(A,C)\rangle$
$+ff$ $+fe$ $+ef$ $+ee$			$-ff$ $-fe$ $-ef$ $-ee$			$+ff$ $+fe$ $+ef$ $+ee$ $-ff$ $-fe$ $-ef$ $-ee$		
$+ef$ $+ee$	✓							
	✓	✓	$-ef$ $-ee$					
	✓	✓		✓				
	✓	✓		✓	✓	$+fe$ $+ee$ $-ff$ $-fe$ $-ef$ $-ee$		
	✓	✓			✓	$+fe$ $+ee$ $-ef$ $-ee$	✓	
	✓	✓		✓	✓		✓	
	✓	✓		✓	✓		✓	✓

(3b) [**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.

SOLUTION:
No. You can't tell because it might not be globally consistent.