ECE 479/579 (Circle your section) **Name:**
Spring 2018 **(please print)**

Final Examination
May 9, 2018

1. (18 points – 3 apiece) Multiple choice questions. Select the answer. **No Partial Credit.**

1.1) Alpha-Beta pruning is useful for

A :	B :	(C:)	D :
Finding solutions to the "block's world"	Finding an optimal subgraph in a and/or graph	Finding optimal decisions in games	Both B & C

1.2) Convert "$\forall y \exists x\ [P(y) \wedge (Q(x) \vee R(x))]$" to clause form

(A :)	B :	C :	D :
$P(y)$ $(Q(f(y)) \vee R(f(y)))$	$P(y)$ $Q(y)$ $R(y)$	$\forall y\ P(y)$ $(Q(A) \vee R(A))$	$\forall y \forall x\ \neg P(y)$ $\vee \neg Q(x)$ $\forall y \forall x\ \neg P(x) \vee \neg R(x)$

1.3) What property of a graph *may* break *any* search for an optimal solution?

(A:)	B:	C:	D:
The graph has negative edge costs	The graph contains cycles	The graph has positive edge costs	Both A & B

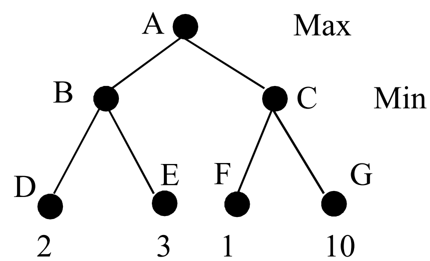
1.4) Regression in Block World planning (the finding of necessary states to carry out a forward rule) is needed for

(A :)	B :	C :	D :
Backward Production System	Resolution	Skolem Function	Forward Chaining

1.5) Provided the rules: frogs are wet; frogs are green; watermelons are green; and wet green things are gross. What can <u>forward chaining</u> discover given X is gross

A :	B :	C :	(D :)
forward chaining does not apply to this type of problem	X could be a frog, a wet watermelon, or a wet green thing	That "gross" extends the types "green" and "wet"	forward chaining discovers nothing new

1.6) For the following game tree, which node will not be explored if alpha-beta pruning were applied (from left-to-right)? PLEASE NOTE: The options are D, C, F and G nodes.



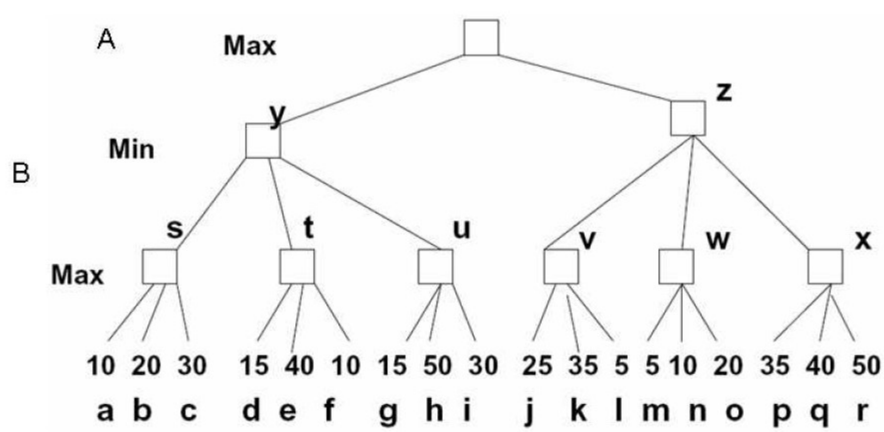
A : **D**	B : **C**	C : **F**	(D :) **G**

2. (12 points – 3 apiece) Please answer True or False. PLEASE NOTE: It can only be True OR False. **No Partial Credit.**

2.1) A heuristic that always evaluates to h(s) = 1 for non-goal search nodes s is always
(F) admissible.

2.2) When using alpha-beta pruning, the computational savings are independent of the order
(F) in which children are expanded.

Based on this tree, using DFS from LEFT to RIGHT to answer 2.3) and 2.4).



2.3) "y" is A's next move.
(T)
2.4) "b" is the node which would not be visited if alpha-beta search is used.
(F)

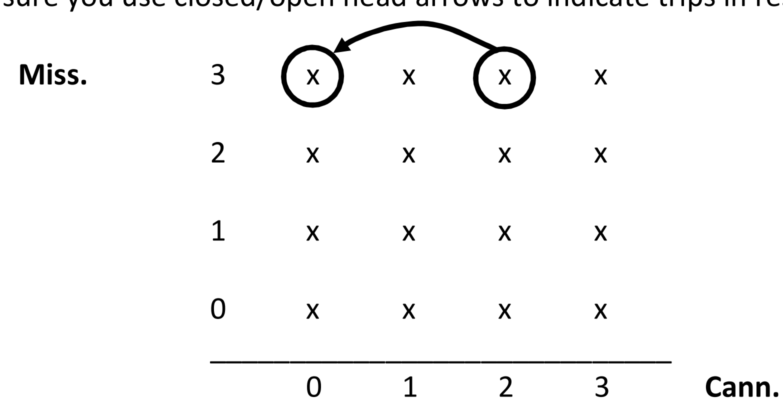
3. (10 points) This question explores the classic problem of transporting three missionaries and three cannibals across a river in a boat that can hold no more than two at a time. The constraint, of course, is that the cannibals cannot outnumber the missionaries anywhere, or else disaster.

To represent a state in the search space, we use a triple (m, c, b) where m is the number of missionaries on the west bank, c is the number of cannibals on the west bank, and b is the position of the boat (w when on the west bank, e otherwise).

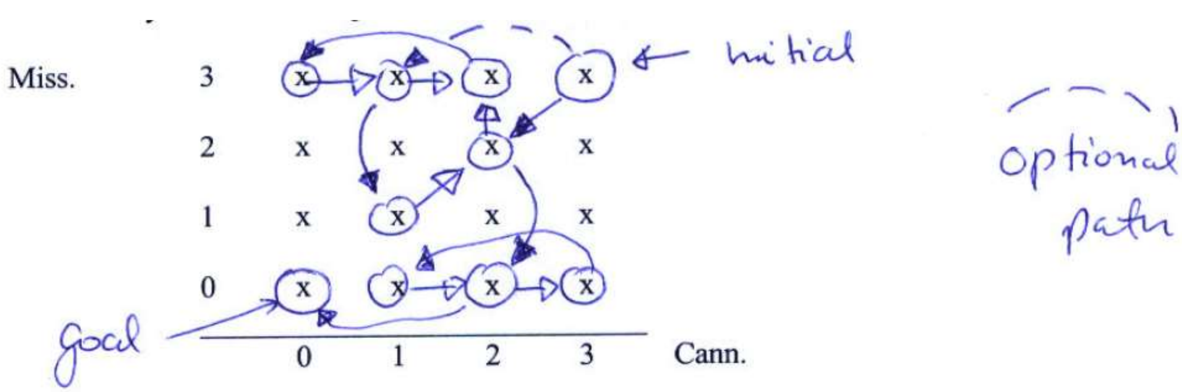
It is useful to make a 4 x 4 diagram in which each point represents the number of missionaries and cannibals on the west bank, leaving the position of the boat ambiguous. West-to-east trips are indicated by arrows with closed heads, while east-to-west trips are indicated by arrows with open heads. For example in the figure below, the closed-head arrow indicates a trip with two cannibals from west to east.

<u>You are to:</u>

1 Identify safe states and circle them on the diagram below.
2 Identify the initial state and the goal state.
3 Find a path that constitutes a solution to this problem. Draw the path on the diagram.
Make sure you use closed/open head arrows to indicate trips in respective directions.



The search space for the missionaries and cannibals.



4. (8 points) Describe the purpose of a Skolem function. Given an example below show how the skolem function would be used.

Before the skolem function:
$\forall x \forall y \forall z \exists w \forall v \big(\mathrm{P}(x, y, z, w, v) \rightarrow \mathrm{Q}(w, v)\big).$

Answer:
Skolem functions are used to remove existential quantification. This is necessary when reducing WFFs with existential quantification to clause form. The skolem function effectively operates by replacing "there exists" with a function of all variables that came before the existential variable.

Example1:
After the skolem function:
$\forall x \forall y \forall z \forall v \big(\mathrm{P}(x, y, z, \mathrm{f}(x, y, z), v) \rightarrow \mathrm{Q}(\mathrm{f}(x, y, z), v)\big).$

5. (8 points) Given the following sentence and its corresponding wff, convert it into clause form.

"For any two wheels, say *x* and *y*, if *x* is propelled by *y*, but it is not in contact with *y*, there must exist some third wheel *z*, such that *x* is propelled by *z* and *z* is propelled by *y*"

Define P(*x,y*) = *x* is propelled by *y*
C(*x,y*) = *x* is in contact with *y*

WFF:

$$\forall x \forall y \left(P(x,y) \wedge \neg C(x,y) \rightarrow \left((\exists z)(P(x,z) \wedge P(z,y))\right)\right)$$

Answer:

Replace $(\exists z)$ and all *z*'s with a Skolem function F(*x,y*)

$$\forall x \forall y \left(P(x,y) \wedge \neg C(x,y) \rightarrow \left(P(x,f(x,y)) \wedge P(f(x,y),y)\right)\right)$$

Dropping the universal quantification and Rewriting the implication (A → B) = ¬A ∨ B

$$\neg\left(P(x,y) \wedge \neg C(x,y)\right) \vee \left(P(x,f(x,y)) \wedge P(f(x,y),y)\right)$$

Applying De Morgan's law:

$$\neg P(x,y) \vee C(x,y) \vee \left(P(x,f(x,y)) \wedge P(f(x,y),y)\right)$$

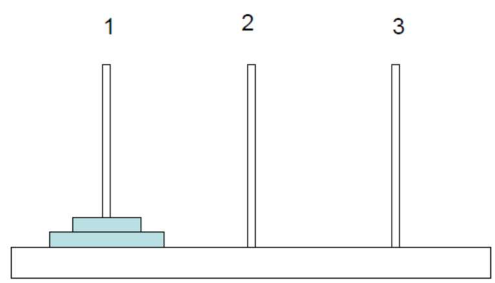
Applying the distributive law:

$$\neg P(x,y) \vee C(x,y) \vee P\left(x,f(x,y)\right)$$
$$\neg P(x,y) \vee C(x,y) \vee P\left(f(x,y),y\right)$$

6. (10 points). Consider an instance of the Towers of Hanoi game. The objective of the game is to move both disks from the left pin to the right one (please see figure), subject to the following constraints:

- Only one disk can be moved at a time,
- The large disk cannot be placed on top of the small disk.

The state of the game can be indicated by a pair (x, y), where x is the pin number for the large disk and y is the pin number for the small disk. For example, (2, 1) is a configuration in which the large disk is on pin 2 and the small one on pin 1.
You are to:
a) Using this notation, list all possible states for our instance for the game (see figure).
b) Create a table indicating which transitions between states are possible in one move.



Answer:

a)

1. (1,1)	4. (2 ,1)	7. (3,1)
2. (1 ,2)	5. (2,2)	8. (3,2)
3. (1,3)	6 . (2,3)	9. (3,3)

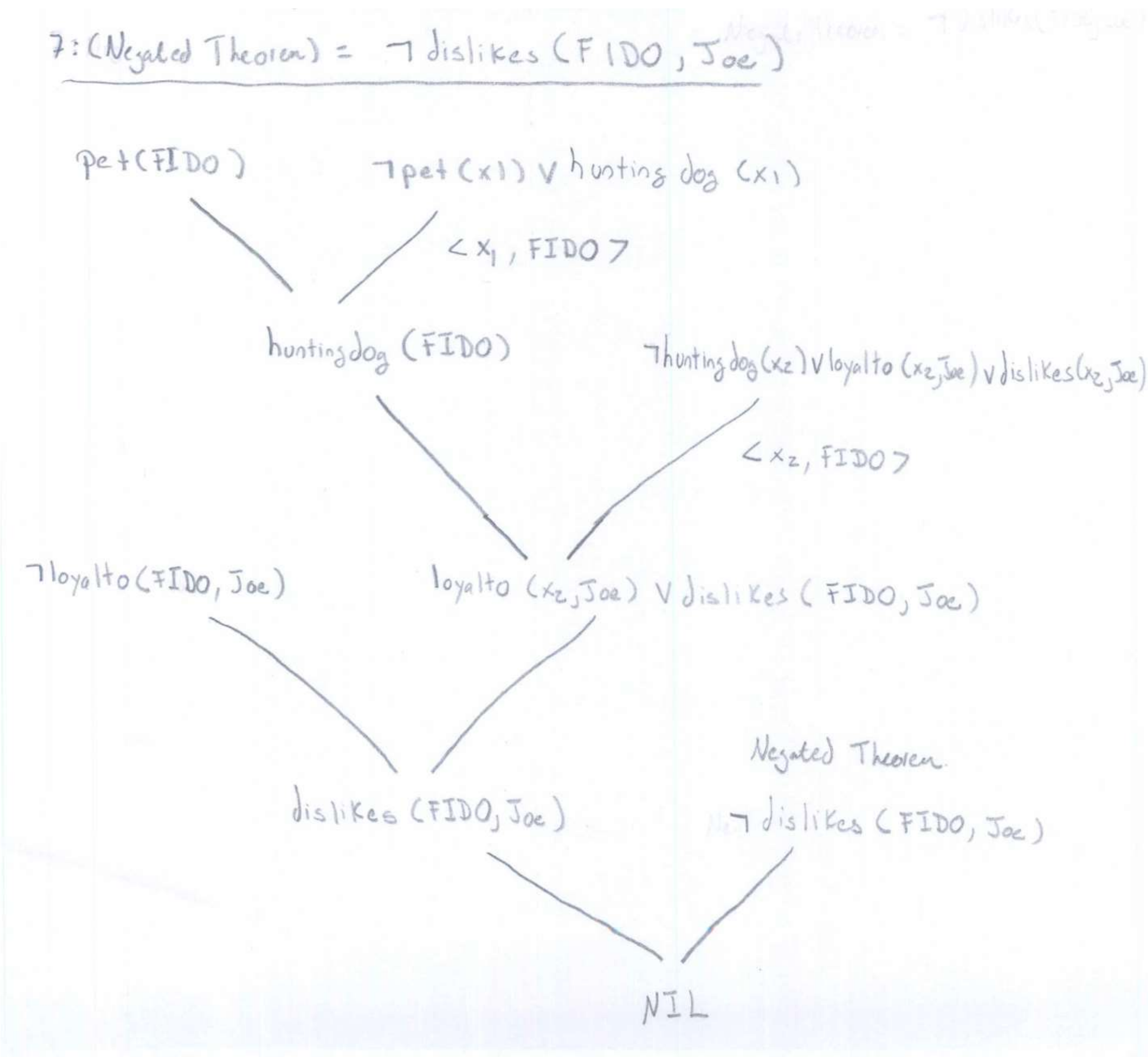
b) State transition table: x denotes a legal transition, for example, from 1 (1,1) to 3 (1,3) (big disc is on peg1, small on peg 3)

	1	2	3	4	5	6	7	8	9
1		x	x						
2	x		x					x	
3	x	x				x			
4					x	x	x		
5				x		x			
6			x	x	x				
7				x				x	x
8		x					x		x
9							x	x	

7. (8 points) Given the following list of axioms:

1. pet(FIDO)
2. ~pet(x1) V huntingdog(x1)
3. master(Joe)
4. ~huntingdog(x2) V loyalto(x2, Joe) V dislikes(x2, Joe)
5. ~loyalto(x2, Joe)
6. trytobite(FIDO, Joe)

Prove by resolution refutation: dislikes(FIDO, Joe).



8. (10 points in TOTAL) Graduate Student Only (Extra Credit for undergraduates)

3A (6 points) Given the following production rules and initial knowledge, execute the rules until there is nothing more to execute. Be sure to show which rule you execute and the facts after each rule is executed.

Initial knowledge:

Android(Marvin)	Brilliant(Marvin)
InvokesPathos(Marvin)	Bored(Marvin)

Rules:

R1: **If** InvokesPathos(x) and PluggedInto(x, y) — **Then** SelfDestruct(y)
R2: **If** Depressed(x) and Brilliant(x) — **Then** Artist(x)
R3: **If** Artist(x) and Depressed(x) — **Then** WroteIHateTheNight(x)
R4: **If** Bored(x) and Brilliant(x) — **Then** Depressed(x)
R5: **If** Android(x) and Bored(x) — **Then** CountElectricSheep(x)

ANSWER:

R4 {x, Marvin} → Depressed(Marvin) + "All of the above knowledge"
R2 {x, Marvin} → Artist(Marvin) + ""
R3 {x, Marvin} → WroteHowIHateTheNight (Marvin) + ""
R5 {x, Marvin} → CountElectricSheep(Marvin) + ""

3B. (4 points) Use regression with the above facts and with R1 refined as follows:
P = {InvokesPathos(x), PluggedInto(x,y)}
D = {PluggedInto(x,y)}
A = {SelfDestruct(y)}

Determine what might cause a vehicle to self-destruct. Be sure to show subgoals and substitutions.

ANSWER – We use regression

The goal condition SelfDestruct(Vehicle), requires that we replace the variable y with Vehicle: {y, Vehicle}.
Replacing y in the precondition produces: InvokesPathos(x) and PluggedInto(x, Vehicle)}.
If we optionally replace x with Marvin, then: InvokesPathos(Marvin) and PluggedInto(Marvin, Vehicle).

Therefore if Marvin was plugged into a vehicle it would self-destruct, or more generally anything that invokes pathos was plugged into the Vehicle.