CSci 4511, Final exam

Instructions: The time limit is 120 minutes. Please write your answers on a separate piece of paper. The exam is open book and notes. You may use electronic devices to ONLY look at either an e-book version or electronic notes. You may not use the internet, program/run code or any other outside resources. (If you are typing on your keyboard/input device for anything other than ctrl-F to find words in the e-book or notes, this is probably not acceptable.) For all questions if you **show work** you are more likely to receive partial credit for incorrect answers. Also support all your answers with clear reasoning.

Problem (1) [10 points] Show the tree of a forward search on this planning problem. Clearly label all the states and actions.
$Action(PutOnUnderwear$,
$Precondition : \neg Wear(Underwear)$
$Effect : Wear(Underwear))$

$Action(PutOnPants$,
$Precondition : Wear(Underwear) \wedge \neg Wear(Pants)$
$Effect : Wear(Pants))$

$Action(PutOnShirt$,
$Precondition : Wear(Underwear)$
$Effect : Wear(Shirt))$

$Action(PutOnShoes$,
$Precondition : Wear(Pants) \wedge \neg Wear(Shoes)$
$Effect : Wear(Shoes))$

$Action(PutOnHat$,
$Precondition : Wear(Underwear) \wedge \neg Wear(Hat)$
$Effect : Wear(Hat))$

$Initial : \neg Wear(Underwear) \wedge \neg Wear(Pants) \wedge \neg Wear(Shirt) \wedge \neg Wear(Shoes) \wedge \neg Wear(Hat)$
$Goal : Wear(Pants) \wedge Wear(Shirt) \wedge Wear(Shoes)$

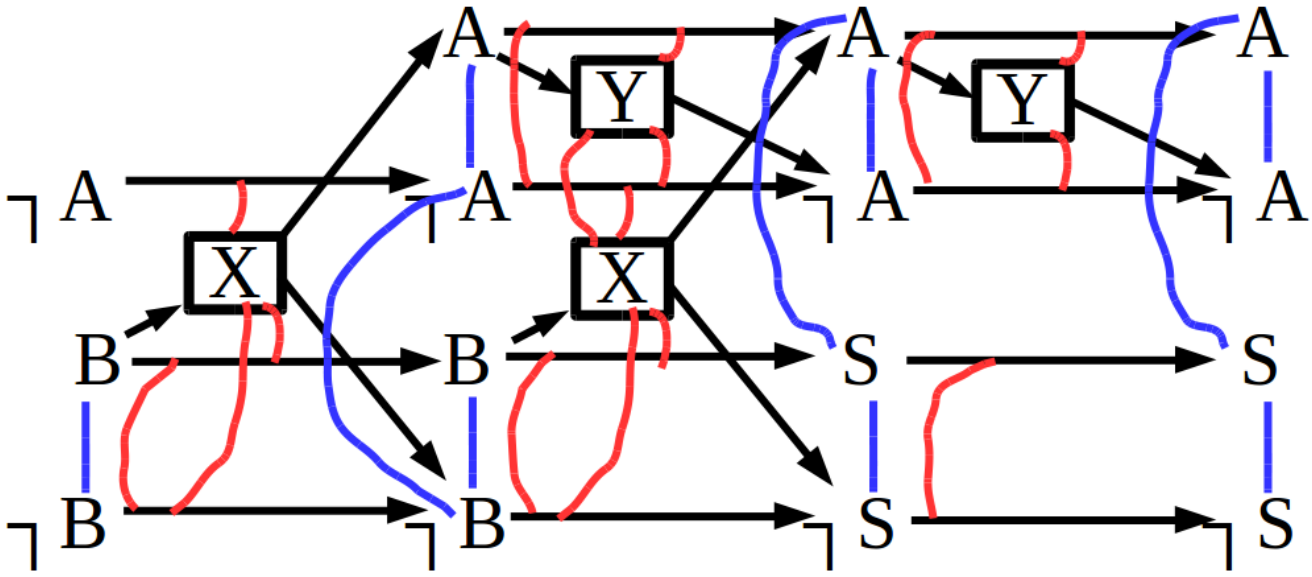
Problem (2) [15 points] For each type of problem, identify the most appropriate method/algorithm to solve the problem. Justify your pick.
(1) You are deep sea explorer. You wan to explore as much of a small cave as you can before you run out of oxygen.
(2) You are taking 30 credits next semester and want to make a program to pick appropriate classes for you.
(3) You are looking for a rare mushroom in the forest. You have a gas analyzer that can find trace amounts of a byproduct of this mushroom and estimate a signal strength.
(4) You are trying to assemble a robot that can both walk and carry things (or stuff!).
(5) You want to build a program that can do your math proofs for you.

Problem (3) [15 points] Answer the following questions. You must explain your reasoning for full credit.
(1) In a general 2-player game theory "game", what is the maximum number of Nash equilibrium for pure strategies? What is the minimum?
(2) Answer part (1), but this time with mixed-strategies.
(3) In a 3-player game theory "game", what is the maximum number of Pareto efficient strategies? What is the minimum?

(MORE ON THE OTHER SIDE)

Problem (4) [15 points] Answer the following true or false questions. In addition to saying whether the statement is true or false, provide justification.
(1) First order logic knowledge bases are always more compact than a corresponding propositional logic knowledge base. That is, you can always express a propositional logic knowledge base in the same or fewer first order logic sentences.
(2) Let x be the number of additional sentences that can be entailed by a knowledge base. Adding more sentences to the knowledge base can only increase x.
(3) Mid-state evaluations can only be applied to zero-sum games.
(4) You can apply minimax to any two player competitive game.
(5) You need to know the range of values in terminal nodes to apply alpha-beta pruning while running minimax.

Problem (5) [15 points] Find 3 **unrelated** errors in the following GraphPlan and your reason for why you think it is wrong. What is shown is supposed to be the entire GraphPlan until convergence.



Problem (6) [15 points] Answer the following questions, and write a short sentence or two for why you think this is sufficient.
(1) Write this sentence into first order logic: "Molly has two or more brothers and exactly one sister."
(2) Explain how you would write this sentence in propositional logic.
(3) How would you write "Molly knows who her brothers are" even if you do not know.

Problem (7) [15 points] For each of the following cases, describe whether you should use: resolution, forward chaining or backward chaining. Also provide the reason why.
(1) When all the sentences are input in a general CNF form.
(2) If your knowledge base is all in the form: $(A_1 \wedge A_2 \wedge A_3 \wedge ... \wedge A_n \Rightarrow B)$. You also will want to run similar queries in the future.
(3) Suppose your knowledge base is very large and filled with only definite sentences. You think only a few sentences are needed for your inference.
(4) Your knowledge base has non-definite sentences, and you think it only takes a few inferences.
(5) When you know nothing about the knowledge base or the query.