6.824 Artificial Intelligence, Fall 1993
Final Exam

name ____

There are five problems. Write your answeres in the space provided or on the back of the sheets. The point value of each problem is shown. You have three hours.

Problem 1. (15 points) This problem is very similar to problem 2 from problem set 2. It is on a Bayesian approach to concept learning. We consider an instance space where each instance assigns Boolean values to n attributes $x_1, \ldots, x_n$. We consider the space of all concepts definable by Boolean expressions over these features. For example, $x_1 \wedge \neg x_2$ is the concept that is true of all instances that assign x_1 the value T and x_2 the value F. Consider the following stochastic grammar for generating Boolean expressions in conjunctive normal form.

$$C \xrightarrow{\frac{1}{2}} (D \wedge C) \qquad C \xrightarrow{\frac{1}{2}} D \qquad\qquad L \xrightarrow{\frac{1}{2}} (\neg A) \qquad L \xrightarrow{\frac{1}{2}} A$$

$$D \xrightarrow{\frac{1}{2}} (L \vee D) \qquad D \xrightarrow{\frac{1}{2}} L \qquad\qquad A \xrightarrow{\frac{1}{n}} x_1 \quad \ldots \quad A \xrightarrow{\frac{1}{n}} x_n$$

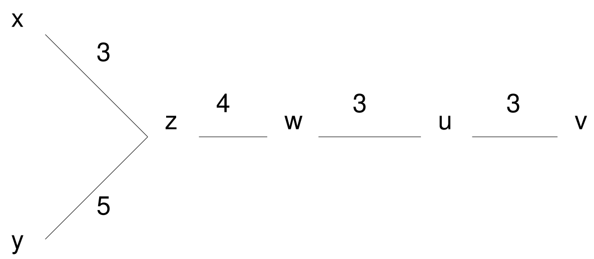
Recall that a stochastic grammar determines a probability distribution over expressions.

Suppose that we are trying to learn a target concept where the target concept has been generated at random using the above grammar. After the target concept has been selected, instances *on which the concept is true* are selected independently at random, giving each instance on which the concept is true equal probability. Let Ψ_1 be the formula $((x_1 \vee x_2) \wedge (x_3 \vee \neg x_4))$ and let Ψ_2 be the formula $((x_1 \vee x_2) \wedge ((x_3 \vee \neg x_4) \wedge (x_5 \vee x_6)))$.

Part a. (5 points) What are the a-priori probabilities of the Boolean expressions Ψ_1 and Ψ_2.

Part b. (10 points) Suppose that we are given a data sequence D consisting of k instances such that both Ψ_1 and Ψ_2 are true of every instance in D. For what values of k and n is $P_r(\Psi_2|D)$ larger than $P_r(\Psi_1|D)$? Justify your answer.

Problem 2. (15 points) This problem is on constraint satisfaction. Consider the graph shown below where each node is a variable and each arc is labeled with a number from 2 to 5. Each node in this graph is a variable that can take on integer values from 1 to 10 (including 1 and 10). Each arc represents a constraint that the two variables connected by that arc must have the same value modulo the number on the arc. For example the arc labeled 3 conecting w and u would be satisfied if w is 5 and u is 8 because both of 5 and 8 are 2 mod 3. Suppose that we know that x is 2, y is 0, and v is 0. Give the variable values that would be infered by value propagation, generalized forward checking, and arc consistency respectively for this graph and these given values for x, y, and v.



Problem 3. (20 points) This problem is on higher order logic and natural language semantics. Consider the determiner "only" as used in the sentences ((only nerds) (watch Star Trek)) or ((some animal in the zoo) (eats (only ants))). The phrases (only nerds) and (only ants) are noun phrases. Give the semantic type for the word only (only maps a noun to a noun phrase). Give a lambda expression for the determiner "only" under which the above sentences have their intuitive meaning.

Problem 4. (25 points) This problem is on the unification algorithm used in lifting. Consider the following inference rules for determining the satisfiability of a set of equality and disequality constraints on expressions. We consider only those expressions constructed from `ONE`, `ZERO`, and the function `CONS`.

```
input rules

1       (NOT (EQUAL ?X ?Y))
        -------------------
        (INPUT! ?X)

2       (NOT (EQUAL ?X ?Y))
        -------------------
        (INPUT! ?Y)

3       (EQUAL ?X ?Y)
        -------------------
        (INPUT! ?X)

4       (EQUAL ?X ?Y)
        -------------------
        (INPUT! ?Y)

5       (INPUT! (CONS ?X ?Y))
        -------------------
        (INPUT! ?X)

6       (INPUT! (CONS ?X ?Y))
        -------------------
        (INPUT! ?Y)

basic equality rules

7       (INPUT! ?X)
        -------------------
        (EQUAL ?X ?X)

8       (EQUAL ?X ?Y)
        -------------------
        (EQUAL ?Y ?X)

9       (EQUAL ?X ?Y)
        (EQUAL ?Y ?Z)
        -------------------
        (EQUAL ?X ?Z)

unification closure rules

10      (EQUAL (CONS ?X1 ?Y1) (CONS ?X2 ?Y2))
        -------------------
        (EQUAL ?X1 ?X2)

11      (EQUAL (CONS ?X1 ?Y1) (CONS ?X2 ?Y2))
        -------------------
        (EQUAL ?Y1 ?Y2)

clash rules

12      (EQUAL ONE ZERO)
        -------------------
        (UNSATISFIABLE)

13      (EQUAL ZERO (CONS ?X ?Y))
        -------------------
        (UNSATISFIABLE)

14      (EQUAL ONE (CONS ?X ?Y))
        -------------------
        (UNSATISFIABLE)

congruence rule

15      (INPUT! (CONS ?X1 ?Y1))
        (INPUT! (CONS ?X2 ?Y2))
        (EQUAL ?X1 ?X2)
        (EQUAL ?Y1 ?Y2)
        -------------------
        (EQUAL (CONS ?X1 ?Y1) (CONS ?X2 ?Y2))

disequation violation rule

16      (EQUAL ?X ?Y)
        (NOT (EQUAL ?X ?Y))
        -------------------
        (UNSATISFIABLE)

occurs-check rule

17      (EQUAL ?X ?Y)
        (OCCURS-IN ?X ?Y)
        -------------------
        (UNSATISFIABLE)
```

Part a. (5 points) Notice that the above rules do not have any way of deriving formulas of the form `(OCCURS-IN` u w`)`. Give an example of an unsatisfiable set of equations and disequations whose unsatisfiability can not be derived by the above rules.

Part b. (8 points) Give inference rules for deriving formulas of the form `(OCCURS-IN` u w`)` such that when your rules are added to the above the result is a correct decision procedure for determining the satisfiability of a set of equality and disequality constraints on expressions, i.e., the rules can derive the formula `(UNSATISFIABLE)` if and only if the input equalities and disequalities are unsatisfiable. This can be done with four rules. Your rules should have the property that forward chaining inference from a finite set of premises can only generate a finite set of assertions.

Part c. (7 points) Let n be the number of expressions in the input equations and disequations (including all subexpressions). Recall that the rules can be run to completion in time proportional to the number of prefix firings of the rules. Give the order of the number of prefix firings as a function of n involved in running the forward chaining unification algorithm defined by the above rules plus the rules you gave in part b. Identify the rule or rules with the largest order of growth in number of prefix firings.

Part d. (5 points) Suppose that we remove rule 15 from the above rule set. Give an example of an unsatisfiable set of equations and disequations such that the above rules together with your rules from part b, but minus rule 15, can not derive the unsatisfiability.

Problem 5. (25 points) This problem is on games. Recall that in α-β search we are only interested in values greater than or equal to α and less than or equal to β. Consider a version of α-β search such that if the static value of a node is at least $\beta+\Delta$ then the procedure terminates and returns β, and if the static value is no greater than $\alpha - \Delta$ then the procedure terminates and returns α. Of course the procedure also terminates if the appropriate depth bound is reached. Consider a game tree in which each max node n has b_1 children with the same static value as n and b_2 children whose static value is the static value of n minus Δ. Dually, a min node m has b_1 children with the same static value as m and b_2 children with static equal to the static value of m plus Δ. Now consider searching this tree to depth d using the version of α-β just described. We assume that the initial values of α and β are $-\infty$ and $+\infty$ respectively and that the root node is a max node with static value 0. We also assume optimal move ordering.

Part a. (5 points) Assume that $b_1 = 2$ and $b_2 = 0$. In this case all static values are 0. In this case how many leaf nodes are there in a four ply search tree under the conditions described above. Explain your answer (do more than simply quote a formula).

Part b. (7 points) For $b_1 = b_2 = 2$ draw the four ply search tree explored by α-β search under the conditions described above.

Part c. (13 points) Give the order of the number of nodes searched as a function of the two branching factors b_1 and b_2 and the depth d. Do not try to compute the exact value — just give the order of growth. Explain your answer.