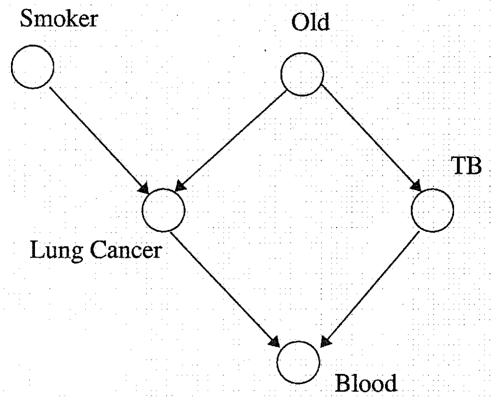
C1 Artificial Intelligence (25 points)

Uncertainty. (17 points) We are given the following Bayes Net and associated probability tables, assuming binary values:



P(Smoker)
0.1

Old	P(TB)
T	0.05
F	0.01

P(Old)
0.2

Smoker	Old	P(Lung Cancer)
T	T	0.3
T	F	0.1
F	T	0.2
F	F	0.01

Lung Cancer	TB	P(Blood)
T	T	1
T	F	0.9
F	T	0.8
F	F	0.01

Compute $P(lungcancer | old \wedge smoker \wedge blood)$. Show all of your work.

Search. (8 points)

Consider the following two heuristic functions for the 8-puzzle problem, both of which are admissible:
h1 = the number of misplaced tiles
h2 = the sum of the distances of the tiles from their goal positions (also called the Manhattan Distance).

You have just been hired by The Best-Darn-Toy Company to provide solutions to hundreds of configurations of the 8-puzzle problem, but there is a big bonus in it for you if you get these solutions done before the competitor company publishes their own set of solutions.

You wisely decide to use Algorithm A* search to find the solutions. Which heuristic should you use in your search? Carefully justify your choice.

C2 Artificial Intelligence (25 points)

Planning. (9 points) Consider that you have a number of trucks that must pick up cargos at various locations and deliver these cargos to a set of destination locations. Trucks may only carry one piece of cargo at a time. There are three unary predicates: *Truck(T)*, *Package(P)* and *Location(L)*. Locations are connected by the binary predicate *Road(L1 , L2)*. Trucks and cargos may be at single locations at a time. This is expressed in the predicate *At(X, L)* where *X* is a truck or a package. Finally we have the predicate *In(P, T)* which says that a package is being carried by a truck and we have the relation *Empty(T)* which says that a truck is empty. Now we consider the actions: *Move(T, L1, L2)* makes a truck travels from *L1* to *L2*, via a road. *Load(T, P, L)* places package onto a truck at a location. *Unload(T, P, L)* removes a package from a truck at a location.

Define the three STRIPS operator schemas for *Move*, *Load*, and *Unload*.

Search. (16 points) Sudoku is a puzzle played on a 9 x 9 grid where, in the initial state, some of the boxes are already filled with a number between 1 and 9 (inclusive). The rules of the game can be summed up as saying “Fill in the grid so that every row, every column, and every major 3x3 block contains the digits 1 through 9”. Below are given two tables (taken from a web source) - the first is showing a Sudoku problem and the second a solution to this problem.

A Sudoku Puzzle

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

A Sudoku Puzzle Solution

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

a) [7 pts] Formalize Sudoku as a constraint satisfaction problem.

b) [9 pts] What is forward checking? What would be the effect of forward checking from the initial state given above on the square in the upper left-hand corner?

C3 Artificial Intelligence (25 points)

Learning. (7 points) Suppose you are running a learning experiment on a new algorithm. You have a data set consisting of 25 examples of each of two classes. You plan to use leave-one-out cross-validation. As a baseline, you run your experimental setup on a simple majority classifier. (A majority classifier is given a set of training data and then always outputs the class that is in the majority in the training set, regardless of the input.) You expect the majority classifier to score about 50% on leave-one-out cross-validation, but to your surprise, it scores zero. Can you explain why?

Planning. (18 points) Assume you have the following 3 operators:

- o1: pre: a
  eff: ¬a ^ b

  i.e., o1 has proposition a as precondition, and its effects are to add b and delete a.

- o2: pre: a ^ c
  eff: ¬a ^ b ^ ¬c

- o3: pre: b ^ c
  eff: ¬c ^ d

Show the first three layers (proposition, action, proposition) of the planning graph starting from an initial state where only a and c are true. Include all the mutual exclusion relations (mutex) and justify each of them with a brief explanation.

C4 Artificial Intelligence (25 points)

Learning. (15 points) AdaBoost is a well-known algorithm for boosting machine learning.

(a) (3 points) Explain the conceptual basis for AdaBoost.

(b) (8 points) Describe in detail the AdaBoost algorithm — be precise, including the formulas for constructing a new model.

(c) (4 points) Describe how a new instance is classified; be precise and include any formulas.

Search. (10 points) Students A, B, C, and D are on the left side of a river and need to reach the right side of the river; it is night and they have only one lantern. The only way across the river is a narrow bridge. Only two people can cross the bridge at once, and it is necessary to carry the lantern to see the way. Thus if two students cross the bridge with the lantern, one must cross back to the left side to return the lantern so that the others can cross the bridge. The four students walk at different speeds — the following is the time that it takes each student to cross the bridge:

```
Student A = 2 minutes
Student B = 3 minutes
Student C = 5 minutes
Student D = 7 minutes
```

When two students cross the bridge together, their time is determined by the slower walker; thus if students A and D cross the bridge together, it takes 7 minutes. The goal is to find a sequence of moves that gets everyone on the right side of the river in the least amount of time, where a move consists of a bridge-crossing (from left to right, or from right to left to return the lantern).

(a) (2 points) What is required for a heuristic to be admissible?

(b) (8 points) Which of the following heuristics are admissible? For each admissible heuristic, prove that it is admissible. For each heuristic that is not admissible, give a counterexample showing that it is not admissible.

h1(state) = the bridge-crossing time of the slowest student who is still on the left side of the river (0 if no students remain on the left side of the river)

h2(state) = the sum of the bridge crossing times of the students who are still on the left side of the river

h3(state) = the difference between the bridge crossing time of the slowest student who is still on the left side of the river and the bridge crossing time of the slowest student who is on the right side of the river