CS 436
Final Exam Solutions

Tuesday December 12, 2006

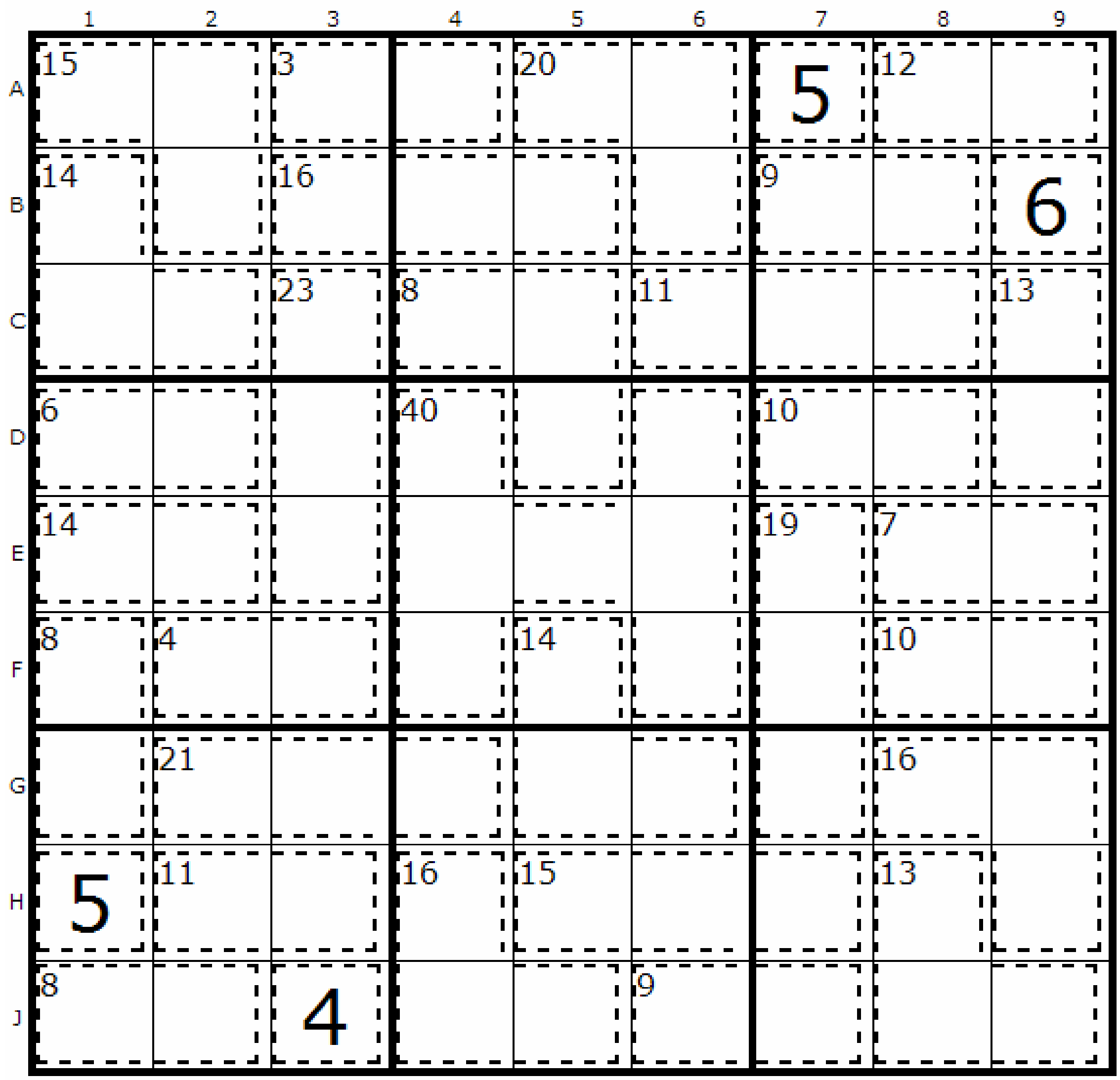
Three pages of notes and calculator allowed.

Name ____

Problem	**Points**	**Max Points**
0	______	**20**
1	______	**20**
2	______	**20**
3	______	**20**
4	______	**20**
5	______	**20**
(Bonus) 6	______	**10**
TOTAL	______	**100 (+ 10)**

0) Killer Sudoku (20 points)

From Sudokulist.org:
"Killer Sudoku is a fiendish variant of Sudoku - the same 9 x 9 board with rows, columns and nine boxes that must be filled in with all the numbers 1 to 9. But instead of seeing some starter clues where some cells are completed for you, in killer sudoku areas of the grid are "caged". Each cage contains a single number which is the sum of all the solutions in that shape. You have to use this information to crack the Sudoku, but all the normal strategies of Sudoku still apply."



(see next page for question)

Explain how you would alter your sudoku solver to solve this new variant Define additional input files and how you would handle the constraints. Make it general enough that it would also handle “killer jigsaw sudoku” as well.

Both regular sudoku and jigsaw sudoku 9 row constraints, 9 column constraints and 9 ‘region’ constraints.

Each contraint is of the form Alldiff(cell1, cell2,.., cell9).

For Killer Sudoku we need to add additional contraints for each ‘caged region’. Given our previous standard of numbering cells left-to-right and top-to-bottom starting at 1 through 81, one of the new caged constraints from the above board is:

Cell(1) + Cell(2) + Cell(11) = 15

So in addition to needing a region files to define regions (which also works for standard sudoku) we need a third file of caged regions. One possible format is :

```
( ((15) 1 2 11)
  ((3) 3 4)
  ((20) 5 6 15)
...
)
```

Once we have read in this file we can use a function which returns the sum of the caged cells then check against the sum.

Note that many killer sudoku sites have a list of sums and possible combinations of numbers to fit the num. It woulkd be wise to consider incoporating such a list in our program for the purposes of using it in our forward checking or ac3 algorithms to eliminate values from the domains of neighboring (in the sense of in the same regions) cells.

1) Machine Reading (20 points)

The time is ripe for the AI community to set its sights on ***Machine Reading,*** the automatic, unsupervised understanding of text.
- Oren Etienne, Michele Banka, and Michael J. Cafarella - 2006

Here we'll think about 'Textual Entailment', which can be thought of as a sub-part of Machine Reading.

S1: WalMart defended itself in court today against claims that its female employees were kept out of jobs in management because they are women.

S2: WalMart was sued for sexual discrimination.

It is obvious to us that S1 entails S2 given a bit of reading comprehension and a few basic external (common sense) facts. However teaching a machine to understand this is hard.

a) What are the unstated facts/relations that logically connect S1 and S2?

These are the main ones:

- *Being sued = defending onself in court (against claims)*
- *Female employees being kept out of some jobs for being female == sexual discrimination*
- *Women are female*

b) What are the necessary steps for us to show logical entailment between S1 and S2 systematically? We did not discuss this in detail in class.. yet if you synthesize your learned knowledge of translating English to logic, add the results of a) and a bit of logical resolution….. you should be able to outline the steps.

1. *Process the sentences*
    a. *'Diagram' (also called part-of-speech-tagging) the sentences using Natural Language processing*
    b. *Extract 'entities' like noun phrases, verb phrases and other liguistic objects from the diagrammed sentence*
2. *Translate the tagged and extracted objects into logical statements*
3. *Form the multiple statements into a KB*
4. *Use standard entailment proof technique to check of the negated version of S2 contradicts the KB (here the KB contains only S1)*

c) What are some problems with turning your outlined steps into an algorithm?

Steps 1 and 2 are quite difficult to do in general. Many AI people are working on this, but it is not yet fundamentally solved.. and it may never be perfectly solved. Yet this is one of those applications of AI where 50% accuracy of translation is a good thing. However if the sentences are translated incorrectly we could output the wrong answer in step 3.

2) Philosophy (20 points)

The Chinese Room

- Human who knows only English; stacks of paper with Chinese symbols; rule book in English, stating which bit of paper to give in response to a given (Chinese) input.
- Human who knows only Chinese on outside of room; passes in Chinese query, receives Chinese response.

While this system would likely pass the Chinese Turing test.. it is clearly problematic. This example is used as a counter argument to the idea that machines can actually think. The argument goes like this:

1. Certain kinds of objects are incapable of conscious understanding (of Chinese).
2. The human, paper and rule book are objects of this kind.
3. If each of a set of objects is incapable of conscious understanding (of Chinese), then any system constructed from the objects is incapable of conscious understanding (of Chinese).
4. Therefore there is no conscious understanding (of Chinese) inside the Chinese room.

a) Who first came up with this example thought experiment?

In 1980, John Searle published "Minds, Brains and Programs" in the journal The Behavioral and Brain Sciences. This paper contains the Chinese Room argument.

b) What step (1-4 above) is problematic? Why?

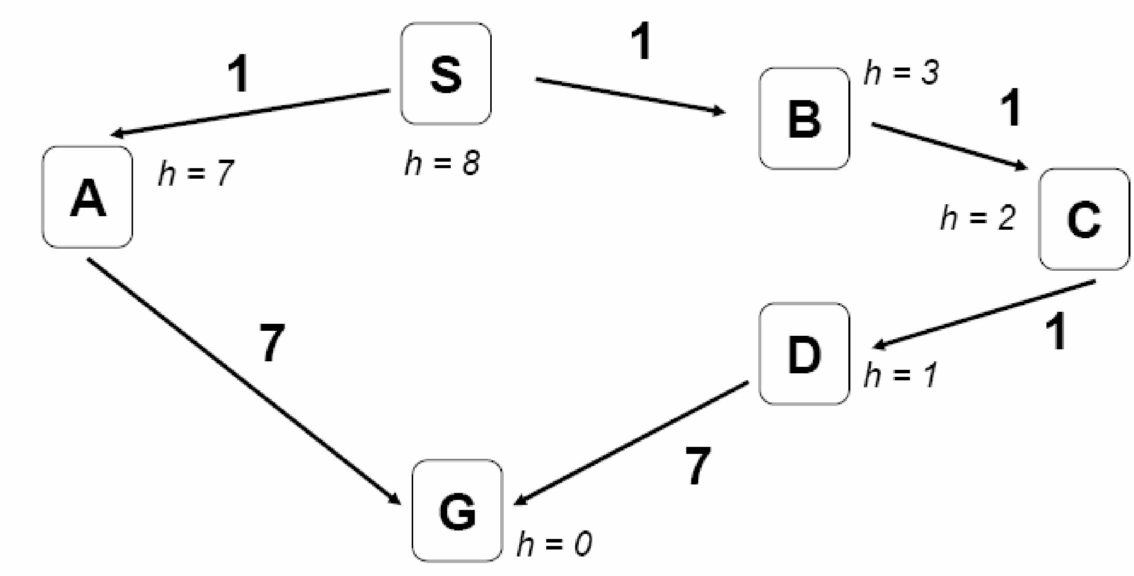
#3 is the main problem. It is open to interpetation.
If you believe it [# 3] and if you believe that humans are composed of molecules then either you must believe either
** That humans are incapable of conscious understanding*
*Or * Individual molecules are capable of conscious understanding.*

c) At a higher level, the 'rule book' plus inputs and actions clearly fit into our basic framework of an 'Agent'. However we are missing a crucial technique in AI, learning. If the agent/human inside of the chinese room had an english-to-chinese dictionary how does this further damage the above argument? What part or parts?

The human inside would be potentially capable of learning Chinese 'by example' (examples are the rule book inputs and outputs) by using the dictionary to look up individual word meanings. Clearly over time the human side would learn Chinese and eventually achieve conscious understanding.

The second key point is that since the rule-book is finite (it fits in the room) it can only answer queries that it has rules for. In this modification the human can demostrate conscious understanding by answering querys not in the rule book.

3) Search Algorithm Understanding



a) Circle the correct termination criteria for A*

1) As soon as it generates a goal state

2) Only when goal state is poped from priority queue

Use the above graph to explain your answer.

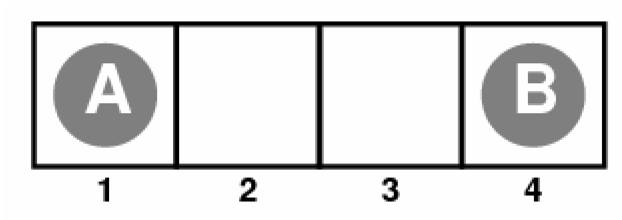
Queue states:

S(8)
B(4), A(8)
C(4), A(8)
D(4), A(8)
A(8), G(10) ← *Option 1: stop here*
G(8), G(10)
Pop G ← *Option 2: stop here*

Obviously if we used option 1 we would not get the optimal solution.

4) Games and Search (20 points)

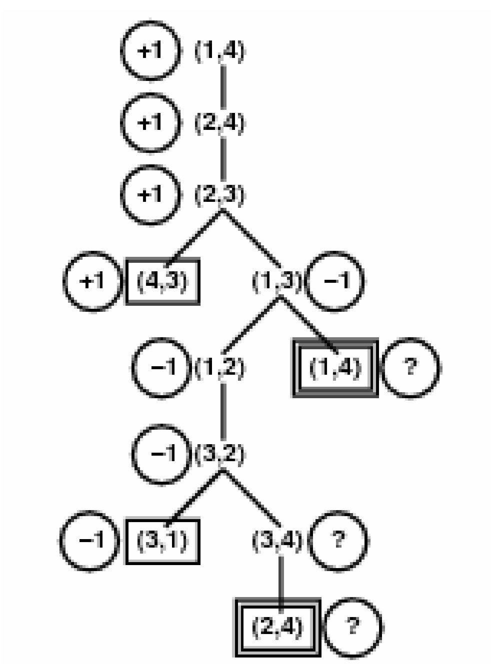
Consider a two-player game featuring a board with four locations, numbered 1 through 4 and arranged in a line. Each player has a single token. Player A starts with his token on space 1, and player B starts with his token on space 4. Player A moves first.



The two players take turns moving, and each player must move his token to an open adjacent space in either direction. If the opponent occupies an adjacent space, then a player may jump over the opponent to the next open space if any. (For example, if A is on 3 and B is on 2, then A may move back to 1.) The game ends when one player reaches the opposite end of the board. If player A reaches space 4 first, then the value of the game is +1; if player B reaches space 1 first, then the value of the game is 1.

(a) On a fresh page, draw the complete game tree, using the following conventions: Write each state as (x,y) where x denotes the location of A, and y denotes the location of B. The state above is **(1,4)**. **Hint: There are only 9 states.**

Put the terminal states in square boxes, and annotate each with its game value in a circle. Put loop states (states that already appear on the path to the root) in double square boxes. Since it is not clear how to assign values to loop states, annotate each with a “?" in a circle.



(b) Now mark each node with its backed-up (the value we return up the tree) minimax value (also in a circle). Explain in words how you handled the "?" values, and why. **Hint: Think about min and max.**

Define:

```
min(-1,?) = -1
max(+1, ?) = +1
```

The "?" values are taken into account by assuming that any choice between a "?" state and winning the game results in chosing the win. The returned value would be "?" in the case of all sucessors having "?" value.

(c) Explain why the standard minimax algorithm would fail on this game tree and briefly sketch how you might fix it, drawing on your answer to (b). Does your modied algorithm give optimal decisions for all games with loops?

Standard minimax would enter an infinite-loop as it is a depth first algorithm. We can fix this by prohibing cycles with a list/stack of visited game states.. we return "?" if the state is already in the list/stack.

In zero-sum games (with no draws) this works fine. However in more general games with draws and wins of differing points/degrees this approach has subtle flaws. Further enhancements to accomidate these games is possible, see game AI literature.

5) Evolutionary Systems (20 points)

(a) Explain the difference between a genotypic representation and a phenotypic representation.

The genotype is the form of the 'DNA" of the GA, ie the bit-string of an individual. The phenotype is the mapping of the bit-string to an evaluation score via the fitness function.

These are generally reguardes as equally important. Selection acts on the phenotype. Mutation and crossover act on the genotype.

(b) Outline the similarities and differences between Genetic Algorithms and Evolutionary Strategies.

GAs have the crossover operator, classic Evolutionary Strategies have only mutation. Classic Evolutionary Strategies operate on a genotype of a list of real numbers (not bitstring), GAs operate on bitstrings. Note that bit-string ES algs are common.

Selection methods can differ slightly also. The GA needs a population of at least two individuals to use crossover.

They similar in that they both have a population of individuals, a genotype and phenotype for each individual, selection and mutation operators and fitness functions… as well as the obvious point that they are both inspired by Darwinian Theory of Evolution.

(c) Outline the simulated annealing cooling schedule, describing the various components.

The cooling schedule of the simulated annealing determines the probability of accepting a 'worsening' move (accepting a poorer fitness individual in a selection step) and this probability changes over time, generally decreasing.

See notes for sample of a basic equation for the cooling schedule.

6) Bonus LISP Review (10 points)

Assume that we have evaluated the following lisp-expression:

(setf a '(3 2 1))

Then, what is the list-structure that results from the following expressions?

(a) (list 'a 1 a)

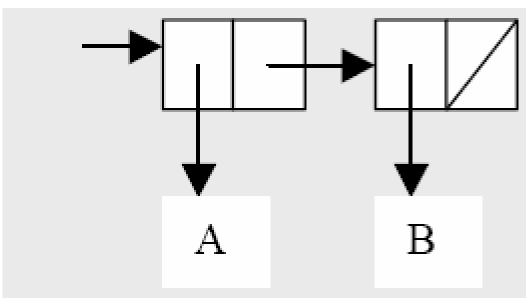
Answer: (a 1 (3 2 1))

(b) (cons a '(a b c))

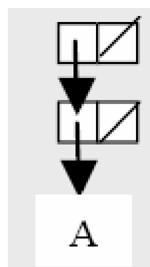
Answer: (a a b c)

Draw Box & Pointer diagram for the following expressions:

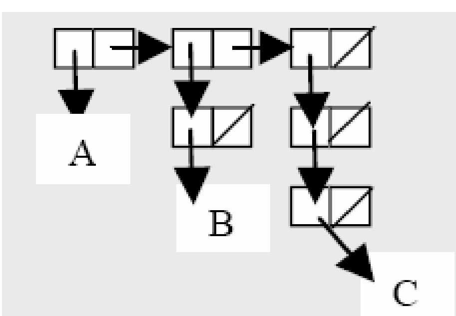
c) (A B)



d) ((A))



e) (A (B) ((C)))



f) (A (B (C)))