6.034 Quiz 1
29 September 2010

Name	
email	

Circle your TA and recitation time **(for 1 point)**, so that we can more easily enter your score in our records and return your quiz to you promptly.

TAs
Martin Couturier
Kenny Donahue
Michael Jagger
Gleb Kuznetsov
Kendra Pugh
Mark Seifter
Yuan Shen

Thu	
Time	Instructor
1-2	Bob Berwick
2-3	Bob Berwick
3-4	Bob Berwick

Fri	
Time	Instructor
1-2	Randall Davis
2-3	Randall Davis
3-4	Randall Davis

Problem number	Maximum	Score	Grader
1	50		
2	50		
Total	100		

There are 11 pages in this quiz, including this one, but not including blank pages and tear-off sheets. Tear-off sheets are provided at the end with duplicate drawings and data. As always, open book, open notes, open just about everything, including a calculator, but no computers.

Problem 1: Rule Systems (50 points)

Engaged in a debate regarding reverse psychology and self-fulfilling prophecy, the 6.034 TAs decide to write a Rule-Based system to determine, once and for all, whether Neo should be able to determine if he is The One in the original film 'The Matrix."

Rules:

P0	IF (OR '(?x) is a noob',
	         '(?x) is disenchanted')
	THEN '(?x) visits The Oracle'

P1	IF '(?x) visits The Oracle'
	THEN '(?x) picks The One'

P2	IF (AND '(?x) is a noob',
	          '(?x) visits The Oracle')
	THEN 'The Oracle says (?x) is not The One',
	     '(?x) is disenchanted'

P3	IF (AND (OR '(?x) is disenchanted',
	              '(?x) is a noob'),
	          'The Oracle says (?x) is not The One')
	THEN '(?x) practices Kung Fu'

P4	IF (AND '(?x) is disenchanted',
	          '(?x) loves (?y)')
	THEN '(?x) proclaims love for (?y)')

P5	IF (AND 'The Oracle says (?x) is not The One',
	          '(?y) proclaims love for (?x)'
	          '(?y) picks The One')
	THEN '(?x) is The One'

Assertions:

A0:	(Neo is a noob)
A1:	(Trinity is disenchanted)
A2:	(Trinity loves Neo)
A3:	(Trinity proclaims love for Neo)

Part A: Backward Chaining (30 points)

Make the following assumptions about backwards chaining:

- The backward chainer tries to find a matching assertion in the list of assertions. If no matching assertion is found, the backward chainer tries to find a rule with a matching consequent. In case none are found, then the backward chainer assumes the hypothesis is false.
- The backward chainer never alters the list of assertions, so it can derive the same result multiple times.
- Rules are tried in the order they appear.

- Antecedents are tried in the order they appear.
- Lazy evaluation/short circuiting is in effect

Neo is the One (30 points)

Simulate backward chaining with the hypothesis

(Neo is the One)

Write all the hypotheses the backward chainer looks for in the database in the order that the hypotheses are looked for. The table has more lines than you need. We recommend that you use the space provided on the next page to draw the goal tree that would be created by backward chaining from this hypothesis. **The goal tree will help us to assign partial credit** in the event you have mistakes on the list.

1 Neo is the One
2
3
4
5
6
7
8
9
10
11
12
13
14
15

Part B: Forward Chaining (20 points)

B1 (5 points)

Does Trinity practice Kung Fu? (does 'Trinity practices Kung Fu' get added to the list of assertions in the forward chainer?)

B2 (10 points)

Let's say, instead, our assertions list looked like this:

A0: (Trinity is a noob)
A1: (Trinity loves Neo)
A2: (Neo is a noob)

Fill out the table below for the first five iterations of forward chaining (matching after 3 is unnecessary)

Iter	Matches	Fire	New Assertions
1			
2			
3			
4	▨		
5	▨		

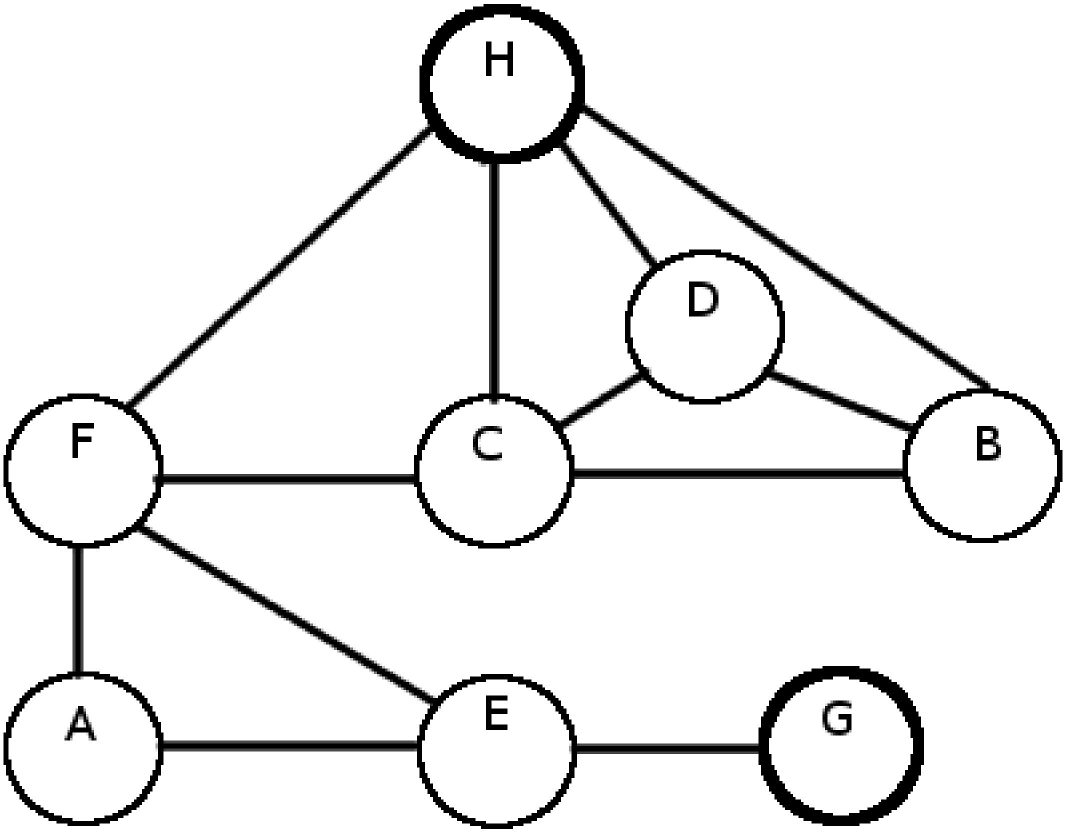
B3 (5 points)

Given these new assertions, what ONE additional assertion do we need to eventually make Trinity also 'The One?'

Problem 2: Search (50 points)

An alien has changed his name to "Ford Prefect" when he came to Earth because he thought he would blend in with the dominant life forms. Ford's friend needs a new name too. Ford suggests he start with "Ford Mustang". The friend decides to use his 6.034 search skills to navigate through a sea of choices to find a good name.

Using a great deal of extraordinarily interesting information deleted from this exam by command of Professor Winston, you find that the friend is faced with the graph shown below.



Part A: Depth First Search + Extended List (14 points)

In all search problems,use alphabetical order to break ties when deciding the priority to use for extending nodes.

Starting at node **H**, find the DFS+ExtendedList path to **G**. Draw the DFS tree and give the final path as well as the list of extended nodes. How many times does your search backtrack? Mark each node as you extend it by putting a number in a circle next to the node corresponding to when it was extended.

Tree:

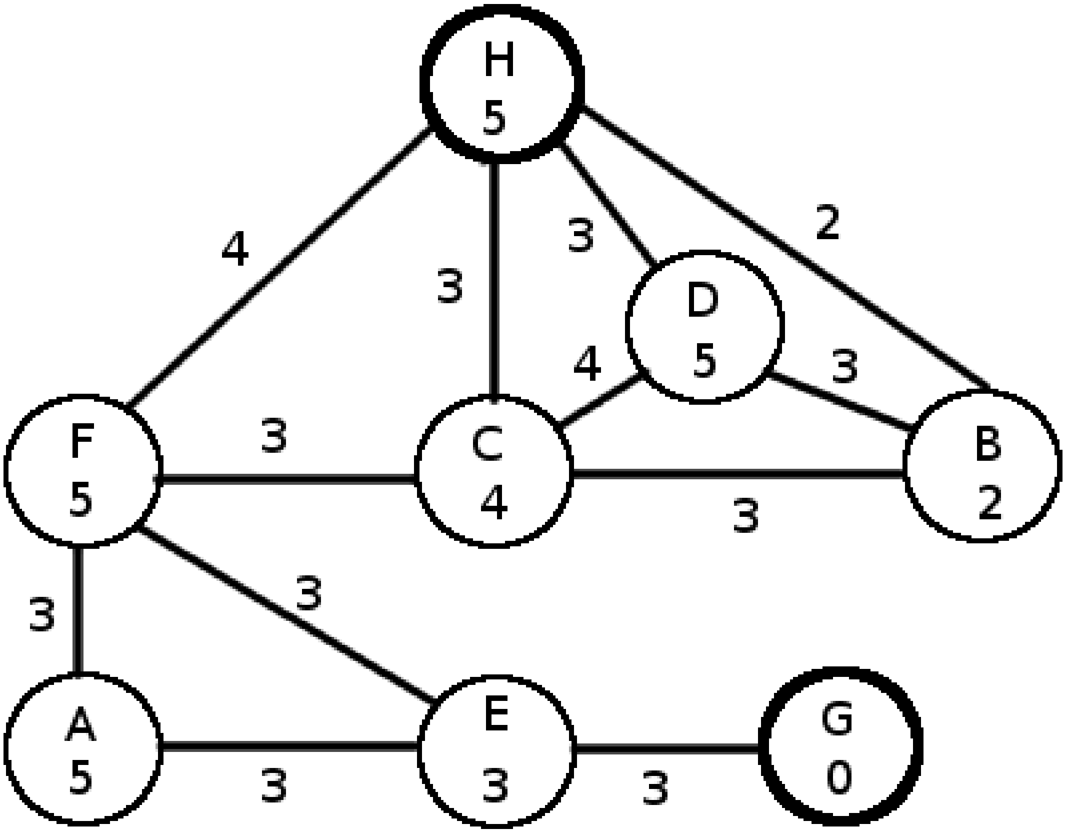
Path:

Extended Nodes in order Extended:

Number of Backtracks:

Part B: Optimal Searches (28 points)

Caaar is convinced that there has to be a better way through the graph. He decides to use additional information to find the shortest path. Distances between nodes are shown on the corresponding edge. Line-of-sight distances to the goal are shown inside each node.



B1 Branch & Bound + Extended List (14 points)

Use B & B search with an Extended List to find both the path and the extended node list. Use this space to draw the corresponding tree and show your work. Mark each node as you extend it by putting a number in a circle next to the node corresponding to when it was extended.

Tree:

Path:

Extended Nodes in order Extended:

B2 A* + Extended List (14 points)

Use the A* search to find both the path and the extended node list. Use this space to draw the corresponding tree and show your work. Mark each node as you extend it by putting a number in a circle next to the node corresponding to when it was extended. **You are to assume that the line-of-sight distances provided are admissible, consistent estimates of the distances to the goal.**

Tree:

Path:

Extended Nodes in order Extended:

Part C: Back to Non-Optimal Search (8 points)

Beam Search (w=2) + Extended List

Now you are to try a Beam Search with a width w=2, with an extended list, **without backtracking**. As before, you are looking for a path from **H** to **G**. Use the line-of-sight-to-goal value provided in part B as your heuristic. Recall that beam search without backtracking is not guaranteed to succeed.

Tree:

Path:

Extended Nodes in order Extended:

Blank page

Tear off sheet, you need not hand this in

Problem 1, Part A

Rules:

P0	IF (OR '(?x) is a noob',
	'(?x) is disenchanted')
	THEN '(?x) visits The Oracle'

P1	IF '(?x) visits The Oracle'
	THEN '(?x) picks The One'

P2	IF (AND '(?x) is a noob',
	'(?x) visits The Oracle')
	THEN 'The Oracle says (?x) is not The One',
	'(?x) is disenchanted'

P3	IF (AND (OR '(?x) is disenchanted',
	'(?x) is a noob'),
	'The Oracle says (?x) is not The One')
	THEN '(?x) practices Kung Fu'

P4	IF (AND '(?x) is disenchanted',
	'(?x) loves (?y)')
	THEN '(?x) proclaims love for (?y)'

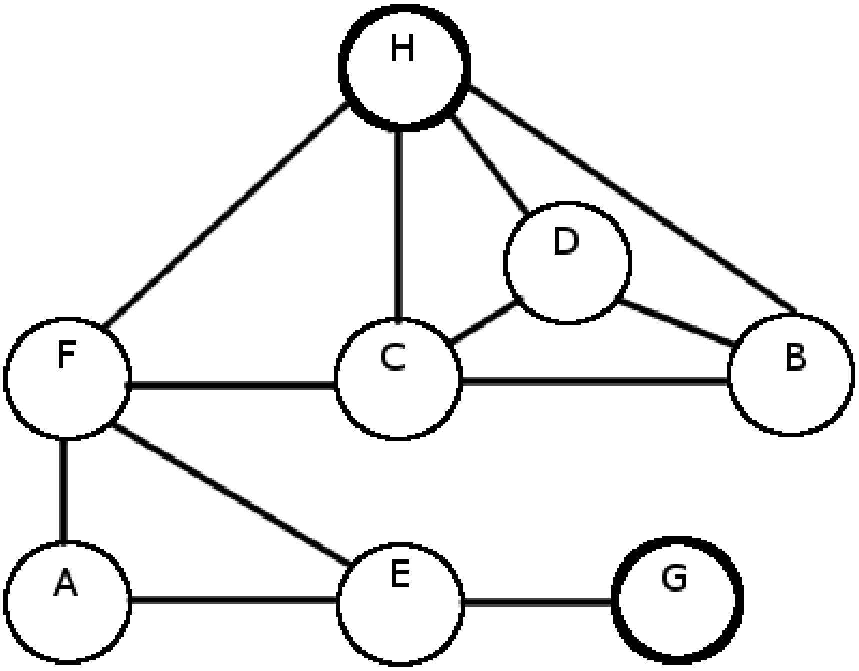
P5	IF (AND 'The Oracle says (?x) is not The One',
	'(?y) proclaims love for (?x)'
	'(?y) picks The One')
	THEN '(?x) is The One'

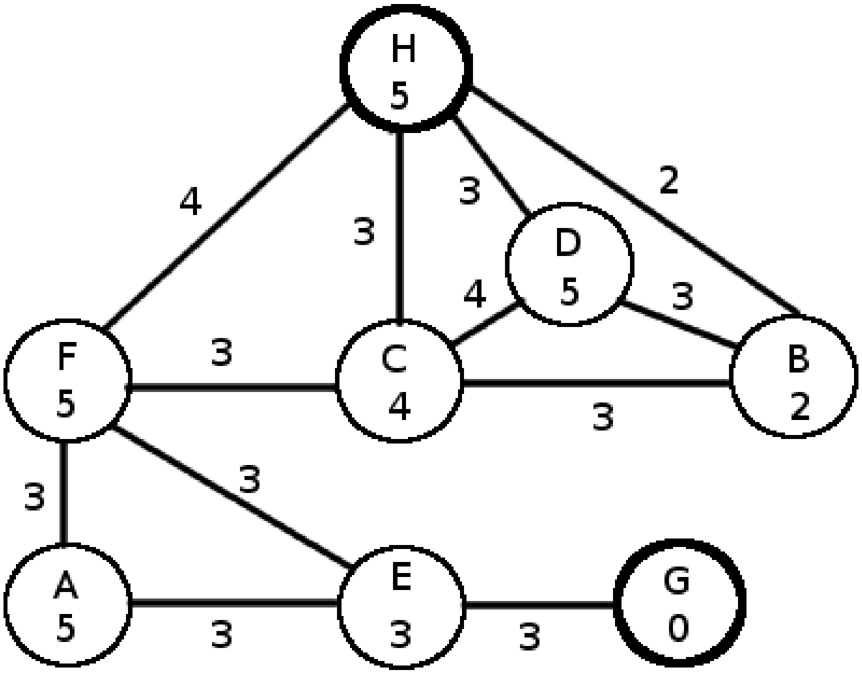
Assertions:
A0: (Neo is a noob)
A1: (Trinity is disenchanted)
A2: (Trinity loves Neo)
A3: (Trinity proclaims love for Neo)

Problem 1, Part B

A0: (Trinity is a noob)
A1: (Trinity loves Neo)
A2: (Neo is a noob)

Problem 2




H
5
4
3
3
2
D
5
4
3
F
5
3
C
4
3
B
2
3
3
A
5
3
E
3
3
G
0