6.034 Quiz 1
September 27, 2006

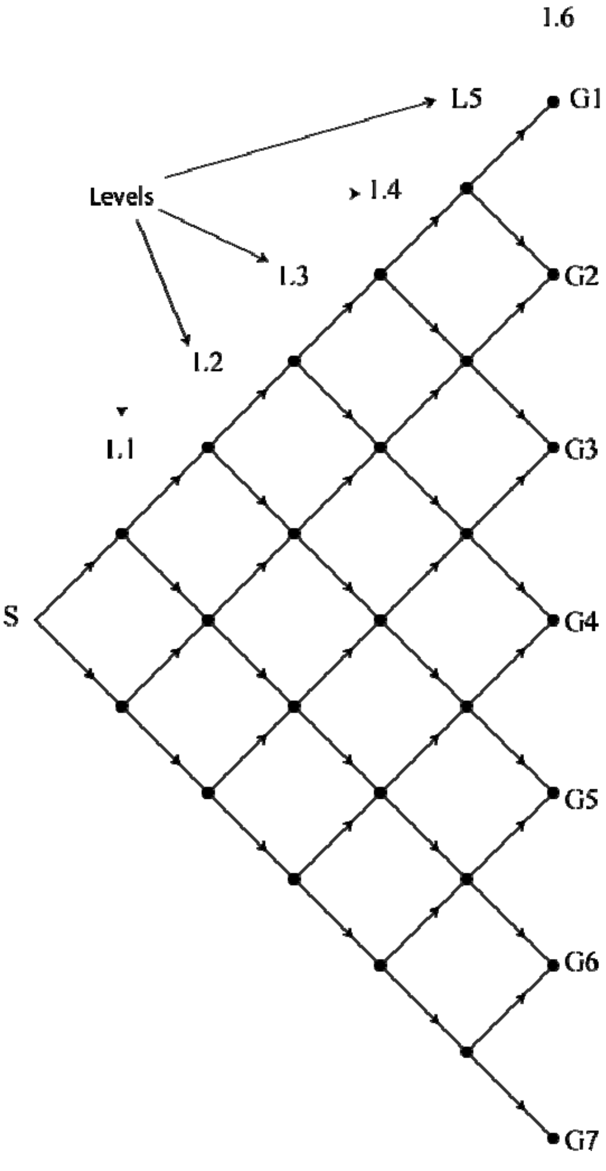
Name	
EMail	

Problem number	Maximum	Score	Grader
1	50		
2	50		
Total	100		

There are 9 pages in this quiz, including this one. A tear-off sheet is provided at the end with duplicate drawings and data.

Problem 1: Search (50 points)

Several 6.034 students are stranded in a strange city at the node marked S. Desperate for a meal, they want to get to any of the restaurants located on the map at the nodes marked G1 through G7. All the streets are one-way streets. Each link between node pairs is 1 unit long.



Part A (10 points)

Stephanie calculates how many paths terminating in one of the goal nodes she would produce if she used the British Museum Algorithm. As expected with the British Museum Algorithm, she **does not use an extended list**. Her result is:

Part B (10 points)

Moira decides to use plain branch and bound, **without** an extended list and with **no estimate** of remaining distance, to compute the shortest path between S and any of the nodes marked with Gs. **If there is ever a choice in which path to extend, she picks the path closest to the top of the page.** Before she can be sure she has the shortest path, she **produces *n* paths starting at S and ending at a node marked G and adds those paths to the queue.** In this case, *n* is

Part C (10 points)

Carlos thinks it would be better to use an estimate of remaining distance with the basic branch and bound procedure, but he still does **not use an extended list**. His estimate for each node is the **horizontal component of the distance** to the nearest node marked with a G. Thus, the distance depends on the layer, as given in the following table:

L1	L2	L3	L4	L5	L6
5√½	4√½	3√½	2√½	√½	0

Is Carlos's heuristic admissible?

If there is ever a choice in which path to extend, he picks the path closest to the top of the page. Before he can be sure he has the shortest path, he produces *n* paths starting at S and ending at a node marked G and adds those paths to the queue. In this case, *n* is

Part D (10 points)

Julian thinks Carlos is on to something, but in his analysis, Julian uses branch and bound together with a different estimate of remaining distance:

L1	L2	L3	L4	L5	L6
5	4	3	2	1	0

Is Julian's heuristic admissible?

Julian still uses **no extended list**. If there is ever a choice in which path to extend, he picks the path closest to the top of the page. Before he can be sure he has the shortest path, he produces *n* paths starting at S and ending at a node marked G and adds those paths to the queue. In this case, *n* is

Part E (10 points)

Kjell thinks it would be a good idea to use branch and bound **with an extended list**, but decides **not to bother with an estimate of remaining distance**.

If there is ever a choice in which path to extend, he picks the path closest to the top of the page. Before he can be sure he has the shortest path, he produces *n* paths starting at S and ending at a node marked G and adds those paths to the queue. In this case, *n* is

Problem 2: Rule systems (50 Points)

Duncan Structivist is a protégé of world-renown architect Frank O. Gehry. In order to win over patrons, Duncan designed a rule-based system which takes descriptions of ordinary buildings and generates new descriptions of similar buildings designed with Gehry's distinctive style. Through careful knowledge engineering, Duncan has provided the system with a way to identify candidates for new Gehry masterpieces.

He created the following rule set:

```
P0:
  (IF ((? X) has non-reflective surfaces)
   THEN (Gehrys (? X) cooks pigeons on the sidewalk))
P1:
  (IF ((? X) has many straight lines)
   THEN ((? X) is functional)
        (Gehrys (? X) has twisted tortured forms))
P2:
  (IF (AND ((? X) is functional)
           ((? X) is not inspiring))
   THEN ((? X) is boring))
P3:
  (IF ((? X) is functional)
   THEN (Gehrys (? X) is inspiring))
P4:
  (IF (Gehrys (? X) has twisted tortured forms)
   THEN (Gehrys (? X) is inspiring))
P5:
  (IF (AND (Gehrys (? X) is inspiring)
           (Gehrys (? X) cooks pigeons on the sidewalk))
   THEN (Gehrys (? X) is a masterpiece))
```

A potential patron approached Duncan anonymously with tentative plans for two new structures. Duncan captured the essentials of the plans in the following data set:

```
A1: (Strata-Center has many straight lines)
A2: (Strata-Center has non-reflective surfaces)
A3: (Pickler-Institute has shiny surfaces)
A4: (Pickler-Institute has many straight lines)
```

Part A: Backward chaining (20 points)

Essential assumptions for backward chaining:

- When working on a hypothesis, the backward chainer first tries to find a matching assertion in the database. If no matching assertion is found, the backward chainer tries to find a rule with a matching consequent. If neither a matching assertion nor a matching consequent is found, the backward chainer assumes the hypothesis is false.
- The backward chainer never alters the database, 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.

Simulate backward chaining with the hypothesis

```
(Gehrys Strata-Center is a masterpiece)
```

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 in the list.

1
2
3
4
5
6
7
8
9
10

The data set (repeated for your convenience):

```
A1: (Strata-Center has many straight lines)
A2: (Strata-Center has non-reflective surfaces)
A3: (Pickler-Institute has shiny surfaces)
A4: (Pickler-Institute has many straight lines)
```

(Gehrys Strata-Center is a masterpiece)

Part B: Forward Chaining (30 points)

Essential assumptions for forward chaining:

- Assume rule-ordering conflict resolution.
- New assertions are added to the bottom of the data set.
- If a particular rule matches assertions in the database in more than one way, the matches are considered in the order corresponding to the top-to-bottom order of the matched assertions. Thus, if a particular rule's has an antecedent that matches both A1 and A2, the match with A1 is considered first.

In order not to distract would-be patrons with unnecessary details, Duncan makes the following change to rule P3:

```
P3:
   (IF ((? x) is functional)
    THEN (Gehrys (? x) is inspiring)
    DELETE ((? x) is functional))
```

Part B.1 (20 points)

Duncan runs a forward chainer on the rules and data set for 1000 steps. In this part of the problem, however, you consider only the first two steps.

The data set is repeated on the next page for your convenience.

In the first step, circle the rules whose antecedents match the initial database:

P0 P1 P2 P3 P4 P5

Also in the first step, circle the rule or rules whose antecedents match the initial database in more than one way.

P0 P1 P2 P3 P4 P5

What new assertion(s) is first added to or deleted from the database in the first step:

In the second step, again circle the rules whose antecedents match the database:

P0 P1 P2 P3 P4 P5

In the second step, circle the rule that actually adds an assertion or deletes an assertion from the database.

P0 P1 P2 P3 P4 P5

What new assertion(s) is added to or deleted from the database in the second step:

Part B.2 (10 points)

After the forward chaining process started in B.1 has finished, has the assertion:

```
(Gehrys (? x) is a masterpiece))
```

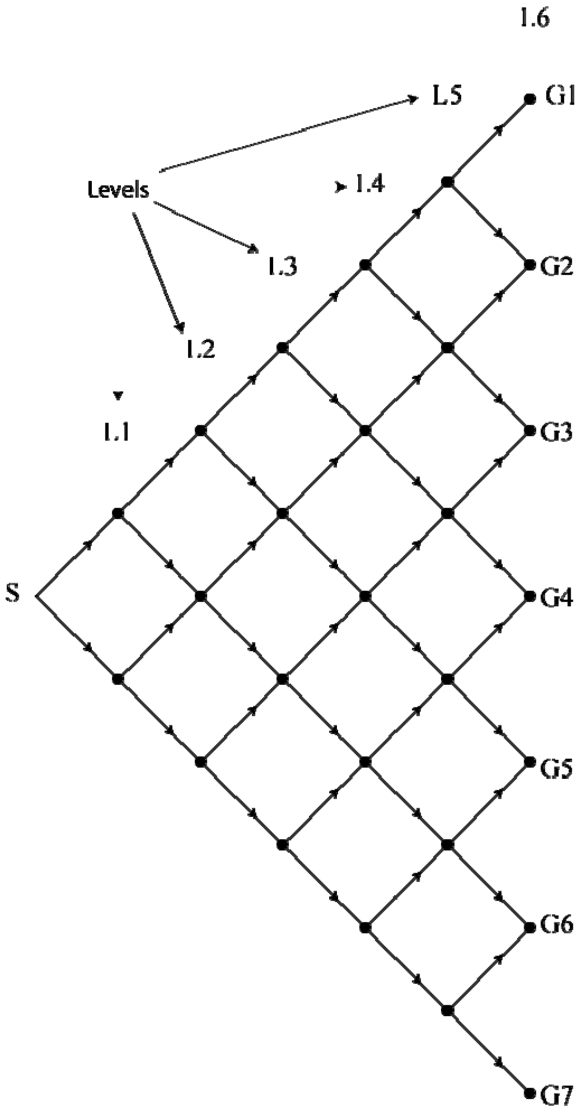
been added to the data set for any binding(s) of the variable x? Provide a brief explanation for your answer. If your answer is yes, be sure to include the bindings(s) of the variable x.

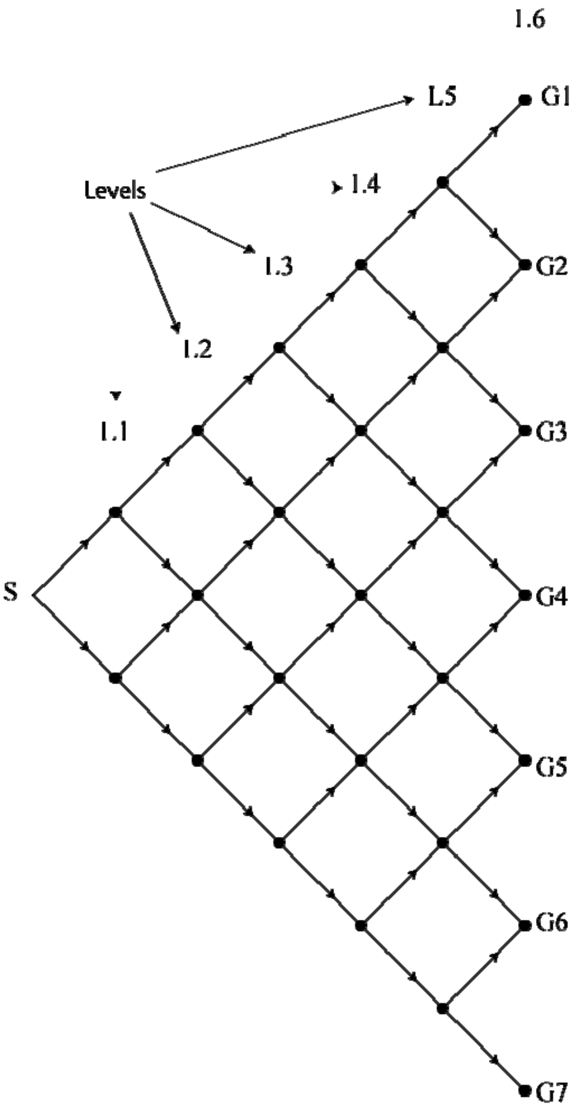
The initial data set (repeated for your convenience):

```
A1: (Strata-Center has many straight lines)
A2: (Strata-Center has non-reflective surfaces)
A3: (Pickler-Institute has shiny surfaces)
A4: (Pickler-Institute has many straight lines)
```

Tear off page

These duplicates provided for your convenience. Nothing on this page will be graded. You need not hand this page in.





These duplicates provided for your convenience. Nothing on this page will be graded. You need not hand this page in.

Backward chaining rules

```
 P0:
   (IF ((? X) has non-reflective surfaces)
    THEN (Gehrys (? X) cooks pigeons on the sidewalk))
 P1:
   (IF ((? X) has many straight lines)
    THEN ((? X) is functional)
         (Gehrys (? X) has twisted tortured forms))
 P2:
   (IF (AND ((? X) is functional)
            ((? X) is not inspiring))
    THEN ((? X) is boring))
 P3:
   (IF ((? X) is functional)
    THEN (Gehrys (? X) is inspiring))
 P4:
   (IF (Gehrys (? X) has twisted tortured forms)
    THEN (Gehrys (? X) is inspiring))
 P5:
   (IF (AND (Gehrys (? X) is inspiring)
            (Gehrys (? X) cooks pigeons on the sidewalk))
    THEN (Gehrys (? X) is a masterpiece))
```

Adjusted rule for use in forward chaining:

```
P3:
   (IF ((? x) is functional)
    THEN (Gehrys (? x) is inspiring)
    DELETE ((? x) is functional))
```

Data set:

```
 A1: (Strata-Center has many straight lines)
 A2: (Strata-Center has non-reflective surfaces)
 A3: (Pickler-Institute has shiny surfaces)
 A4: (Pickler-Institute has many straight lines)
```