CS322 Fall 1997
Assignment 5

Due: 1:30pm, Friday 10 October 1997.

Question 1

In this question you will look at backtracking versus arc consistency for solving CSP problems.

Consider a scheduling problem, where there are 5 activities to be scheduled in 4 time slots. Lets represent the activities by the variables *A*, *B*, *C*, *D*, and *E*, and suppose the domain of each variable is $\{1, 2, 3, 4\}$ and the constraints are: $A > D$, $D > E$, $C \neq A$, $C > E$, $C \neq D$, $B \geq A$, $B \neq C$, $C \neq D + 1$.

[Before you start this, try to find the legal schedule(s) using your own intutions.]

6 marks Show how backtracking can be used to solve this problem. To do this you should draw the search tree generated to find all answers. Indicate clearly the valid schedule(s).

6 marks Show how arc consistency can be used to solve this problem. To do this you need to

- draw the constraint graph,
- show which elements of a domain are deleted at each step, and which arc is responsible for removing the element,
- show explicitly the constraint graph after arc consistency has stopped.
- show how splitting a domain can be used to sove this problem.

Question 2

In this question you will implement the graph searcher for finding the shortest video presentations for the previous two assigments.

For example, given the segment database (from Assignment 3),

```
% segment(SegId,Time,Topics) is true if segment SegId takes time Time
% and covers the topics in Topics.
segment(seg0,10,[welcome]).
segment(seg1,30,[skiing,views]).
segment(seg2,50,[welcome,computational_intelligence,robots]).
segment(seg3,40,[graphics,dragons]).
segment(seg4,50,[skiing,robots]).
```

you also need to assume that you have a list of all segments:

```
% all_segments(L) is true if L is the list of all segments.
all_segments([seg0,seg1,seg2,seg3,seg4]).
```

Suppose that a node is represented as $vp(To_Cover, Segs)$ where *Segs* is a list of segments that must be in the presentation, and *To_Cover* is a list of topics that also must be covered, such that none of the segments in *Segs* covers any of the topics in *To_Cover*.

Write the *neighbors* relation, so that

```
ask neighbours(vp([welcome,robots],[]),Neighs).
```

has one answer (perhaps in the other order):

```
Answer: neighbours(vp([welcome,robots],[]),
                   [vp([],[seg2]),vp([robots],[seg0])]).
```

You also need to define the *cost* predicate, where the cost of an arc is the time of the segment added, and an *is_goal* predicate.

Show how this can be used with the *hsearch* program (available from `~cs322/cilog/hsearch.pl` and from the web page, where `graph.pl` and `graph2.pl` are axiomatizations of the graphs in the book). The following query should give the shortest presentation:

```
ask hsearch(shortest,[node(vp([welcome,skiing,robots],[]),[],0,0)],P).
```

Show that it works. (Note that asking "how" will show the steps of the search).

Question 3

For each of the questions in this assignment, please estimate the amount of time you spent on it. Was this a reasonable amount of time for one week in one course?