CS533 — Written Homework 2 — Due Friday, November 3

1. Consider the following gear-box design problem. The input-to-output ratio should be 63/16. The dimensions of the box are 2 inches square with the lower left corner at the origin. The input shaft should be centered at (0.8, 0.8) and the output shaft should be centered at (1.7, 0.3). Assume the radius of each shaft is 0.125 and that each shaft passes completely through the gear box. Use the gears listed on the slide distributed in class. Hint: I believe a OneTwoOne design is required.

2. Write a rule that would test for a collision between a gear and a shaft and mark the shaft as being in an illegal position. Note, if you need to use two parameters of the same type in the condition, you can use the syntax NEEDS(type, name1) and NEEDS(type,name2) to give them two different names (name1 and name2). Assume that the radius of each shaft is 0.125 inches.

3. Consider the following design problem (modified from Gonzalez & Dankel, 1995): Suppose that John Smith, a student, is changing apartments and needs to move his belongings (listed in the table below). He must pack them into boxes subject to the following constraints.

- The total number of boxes should be minimized (at least approximately).
- All fragile items must be padded.
- If a box contains a fragile item, then it cannot contain any objects heavier than that item.
- The sum of the sizes of the objects in the box cannot exceed 10 units.
- Every item must be packed in some box.

Item	Count	Weight	Size	Fragile?
TV	1	heavy	10	yes
Books	7	heavy	2	no
VCR	1	medium	4	no
Lamp	2	medium	8	yes
Set of Dishes	1	medium	5	yes
Set of Glasses	1	medium	4	yes
Coat	2	light	3	no
Jeans	4	light	2	no
shirts	8	light	1	no

(a) Analyze each of the constraints in terms of their degree of locality.

(b) Assume that a state is represented by two items: (i) list of boxes where box is a list of objects and each object can be padded or unpadded and (ii) a list of objects remaining to be boxed. Write C++ class definitions to represent such states. Suggestion: represent each object separately (i.e., there are 7 books, so there should be 7 distinct book objects).

(c) Write a C++ function to determine whether a state is a feasible solution.

(d) Describe a constructive problem space for this problem. Describe your operators in pseudocode. The problem space should not do any constraint propagation.

(e) For each of the constraints given above, indicate whether it can be used to propagate information. If so, write a pseudocode description of a rule that would propagate this information.