Homework 4, due midnight Sunday Apr 22

Full electronic submission due then, paper version due Friday in TA mailbox (101A Smith). Ten percent late penalty immediately, and for each following 24 hours.

Working together

- A&S

Do not submit problems in the above section.

Working alone: for submission

You may **talk** with another student and **share ideas** with another student for these problems, but you MAY NOT look at another student's answers or code for any reason (not even debugging - but use the intepreter!). You may of course discuss anything with the TA or professor.

Pay special attention to what is required. If the question asks for drawings, process illustrations, answers, descriptions, etc., be sure to provide those things.

1. Consider the following set of bids, where each pair represents `(price . quality)`.

   $$((10.5)(15.10)(20.20)(5.2)(30.21))$$

   Assume your user has a utility[1] function as follows:

   $$Utility_{bid} = f(quality_{bid}) - g(price_{bid})$$

   A basket of bids can be either a single bid, or a list of bids (read that sentence again). A bid may only appear once in a basket. The price of a basket is the sum of all bid prices, and the quality is the sum of all bid qualities. The utility of a basket is

   $$Utility_{basket} = f(quality_{basket}) - g(price_{basket})$$

   Write a function that returns the best basket (highest utility) for the three parameters f,g, and some maximum budget. A basket that exceeds the budget has undefined utility, *(not zero)*. Use your function make-combos from the previous homework to find the basket of bids that has the highest utility when:

   (a) $f = 2 * q$, $g = .01 * p^2$, and the user has $10;
   (b) $f = 2 * q$, $g = .01 * p^2$, and the user has $100;
   (c) $f = q^2$, $g = 3 * p$, and the user has $10;
   (d) $f = q^2$, $g = 3 * p$, and the user has $50;
   (e) $f = q^2$, $g = 3 * p$, and the user has $100;

2. Review the message-passing section of the textbook. Write code similar to that used in class to write the types **ordered set** and **unordered set**. Each type will be able to perform **adjoin** (only for Scheme primitive numbers which are not objects) and **union** of either kind of set. Ord-ord union must return a type ord object.

3. The type coercion system we discussed in class will help control the need for an exponential number of type columns in a data-directed system. However, for now assume we **do not** have such a system. Write packages for ordered sets and for tree sets. Have the ordered package install adjoin (only for Scheme primitive numbers) and union for type combinations where 'ord comes first (i.e. (ord ord) and (ord tree)). Have the tree package install adjoin and the other unions. Demonstrate all by writing generic functions that call apply-generic to accomplish their task.

[1] Utility is a word coined by Bernoulli in 1738 to describe how "useful" or "good" a solution is to an agent (a person, a program, etc.). For example, in this utility function a Toyota with price 20,000 and quality 10,000 may or may not have a higher utility than a house with price 200,000 and quality 100,000, depending on procedures f and g.