

Homework 1: Introduction to Programming Concepts

Due: normal problems on Tuesday, August 29, 2006; extra credit problems on August 31, 2006. In this homework you will learn some of the basics of Oz and the Mozart system, and will get an overview of programming concepts.

For all programming tasks, you must run your code using the Mozart/Oz system.

You must also provide evidence that your program is correct (for example, test cases). Hand in a printout of your code and the output of your testing, for all questions that require code. For testing you may find the following procedure helpful.

```
proc {Assert B Msg}
  if B then skip
  else raise "Assertion Failed: " # Msg end
end
end
```

The `Assert` procedure can be used as in the following statement.

```
{Assert ({Comb J I} == {CombB J I}) J#I }
```

Note that you would not use `Browse` around a call to `Assert`.

Be sure to clearly label what problem each function solves with a comment.

Don't hesitate to contact the staff if you are stuck at some point.

Read Chapter 1 of the textbook [RH04] You may also want to refer to reference and tutorial material on the Mozart/Oz web site <http://www.mozart-oz.org/>.

Textbook Problems

The following problems are from the textbook [RH04, section 1.18].

1. Select enough of the following to achieve a total of at least 30 possible points.
 - (a) (20 points) Problem 2, calculating combinations.
 - (b) (10 points) Problem 3, program correctness.
 - (c) (10 points) Problem 5, lazy evaluation
 - (d) (10 points) Problem 7, explicit state.
 - (e) (10 points) Problem 8, explicit state and functions.
 - (f) (15 points) Problem 10, explicit state and concurrency.
2. (extra credit) Do as many of the above selected problems as you find interesting. The extra credit points you receive will be based on the possible points noted above.
3. Other extra credit problems from the textbook.
 - (a) (20 points; extra credit) Do Problem 6, higher-order programming.
 - (b) (30 points; extra credit) Do Problem 9, memory store.

Other Problems

4. (20 points; extra credit) Write a class `Point` that implements 2D points with operations `Move`, `GetX`, and `GetY`.
5. (50 points total; extra credit) Read a paper on one of the following topics:
 - expressive power [Fel90, Fel91, Mit91, Mit93], or
 - paradigms of programming [Flo79].

You can also find some other published research article in a journal or conference proceedings on these topics. (By a published research article, I mean an article that is not in a trade journal (e.g., it has references at the end), and that is from a refereed journal or conference. *Publication* means the article actually appeared in print (or an on-line refereed venue), and was not just submitted somewhere. So beware of technical reports on the web. It's okay to get a copy of a published article from the web, although I highly encourage you to physically go to the library.)

Write a short (1 or 2 page maximum) review of the article, stating:

- (10 points) what the problem was that the article was claiming to solve,
- (20 points) the main points made in the article and what you learned from it,
- (20 points) what contribution it make vs. any related work mentioned in the article.

In your writing, be sure to digest the material; that is, don't just select various quotes from the article and string them together, instead, really summarize it. If you quote any text from the paper, be sure to mark the quotations with quotation marks (“ and ”) and give the page number(s).

If you do a different article than one of those mentioned above, then hand in a copy of the article with your review. In any case, be sure to explicitly cite the paper you reviewed.

References

- [Fel90] Matthias Felleisen. On the expressive power of programming languages. In N. Jones, editor, *ESOP '90 3rd European Symposium on Programming, Copenhagen, Denmark*, volume 432 of *Lecture Notes in Computer Science*, pages 134–151. Springer-Verlag, New York, NY, May 1990.
- [Fel91] Matthias Felleisen. On the expressive power of programming languages. *Science of Computer Programming*, 17(1-3):35–75, December 1991.
- [Flo79] Robert W. Floyd. The paradigms of programming. *Communications of the ACM*, 22(8):455–460, August 1979.
- [Mit91] John C. Mitchell. On abstraction and the expressive power of programming languages. In *Conference on Theoretical Aspects of Computer Software, Sendi Japan*, September 1991.
- [Mit93] John C. Mitchell. On abstraction and the expressive power of programming languages. *Science of Computer Programming*, 21(2):141–163, October 1993.
- [RH04] Peter Van Roy and Seif Haridi. *Concepts, Techniques, and Models of Computer Programming*. The MIT Press, Cambridge, Mass., 2004.