CSci 658-01: Software Language Engineering
Spring 2018 Assignment #3

H. Conrad Cunningham

5 March 2018

Revised Deadline Monday, 2 April, 2018, 11:59 p.m.

Original deadline Thursday, 29 March, 2018, 11:59 p.m.

General Instructions

All homework and programming exercises must be prepared in accordance with the instructions given in the Syllabus. Each assignment must be submitted to your instructor by its stated deadline.

Assignment #3 Description

This assignment may overlap with Assignment #4, which likely will be a DSL analysis and design exercise.

Alternative A (State Machine Internal DSL)

Unless you chose an alternative problem, Assignments #1 and #2 required you to develop a *semantic model* and an *external DSL* for Fowler's State Machine case study. You could use Python 3 or some other appropriate language (excluding Java or Scala for which we had solutions). You could use any of the approaches in the class materials for designing and implementing external DSLs.

This alternative for Assignment #3 asks you to design and implement an appropriate *internal DSL* for the State Machine problem in the same programming language. You may use the semantic model you developed for Assignment #1.

You may use any of the internal DSL techniques we have talked about in class or that are in the Fowler DSL book. Be systematic in your use of the DSL techniques. That is, identify what techniques you are using and, if appropriate,

how you are adapting the technique to the programming language, problem, and DSL design. Think about design patterns and good programmingn practices.

Alternative B (Lair Configuration DSL)

This alternative for Assignment #3 asks you to design and implement *one* of the following options for the Lair Configuration case study. As above, approach the task in a systematic manner. You may use the semantic model I provided in Python 3.

- a Python 3 *internal DSL* for which I did **not** provide a solution (e.g., Literal Collection, Nested Function calls, etc).
- a Python 3 *external DSL* using a different approach than I did for Parsita-based DSL. (Use either a different parsing technique or a significanly different syntax).

Alternative C (Sandwich DSL)

This alternative for Assignment #3 asks you to design and implement *one* of the following options related to the Sandwich DSL case study. As above, approach the task in a systematic manner. You will need to implement an appropriate semantic model.

- a Python 3 *internal* DSL
- a Python 3 *external* DSL

The class lecture notes give solutions in Haskell, Scala, and Lua, with the older Lua problem description being slightly different.

Alternative D (Choose Your Own Adventure)

Investigate use of the following:

- a Python 3 parser generator library (e.g., Arpeggio/textX)
- a different parser combinator library (e.g., Parsy) than the one I used.

Then design and implement an *external* DSL for the (a) State Machine, (b) Lair Configuration, or (c) Sandwich DSL case studies. For the State Machine you may use your semantic model from Assignments 1 and 2. For the Lair Configuration, you may use my semantic model from class.

For All Alternatives

- When complete, submit your source code, test DSL input, test program, etc. files to Blackboard.
- Be sure to describe your DSL's syntax and give instructions on how to build your program from the source code.
- Remember that the instructor prefers to compile and execute your solution on his MacOS systems. Make it easy for him by identifying what software is needed, etc.
- This is an individual assignment. Document what resources you used in constructing your solution.

Looking Ahead

- Assignment #4 will likely be a group DSL analysis and design exercise for a problem. (I am considering using a text-based, choose-your-own-adventure game as the problem.)
- Finally, you will need to complete a project in which you choose your own problem and then design an implement a DSL to solve it. You are encouraged to choose something of interest to your research or professional goals or something else interesting.