Growing Your Language

CS320 Programming Languages Midterm Exam (Due: May 15, 2020)

1 OVERVIEW

The following is the problem statement of the midterm exam:

> "Extend a language from the course to include a new language feature."

You should choose a language feature that has not been covered by the course, and extend a language you have learned from the course so that it includes the feature. The goal of the exam is to write an essay answering the problem and to implement an interpreter of a language that you have defined in the essay. You should write the essay in English and implement the interpreter in Scala. Section 2 gives more details about what you should do.

The exam is a take-home open-book exam. You may refer to the lecture notes and the course reading materials. In addition, you can search online in order to find a new language feature. However, you cannot get any help from others. You must write the essay and implement the interpreter solely by yourself.

You must submit the honor code for the course as well as the essay and the source code of the interpreter. You should sign the honor code before submitting it. To check whether you violate the honor code, **Professor Sukyoung Ryu and TAs** will interview randomly selected students after grading submissions. **Cheating is strongly forbidden. Cheating will get you an F.**

2 INSTRUCTIONS

2.1 Watching a Video

- Watch the "Growing a Language" video by Guy L. Steele Jr. The script of the video is available.
- You must summarize the content of the video in the essay.
- You must include your own opinions on the video in the essay.

2.2 Choosing a Feature

- Choose a real-world language.
- Find a feature from the language. The feature must not have been covered by the lectures, the exercises, and the project. We disallow you to choose static type checking as a feature since it will be covered after the midterm exam. You may choose one of the following examples:
  - arrays (mutable and random-accessible)
  - exceptions and handlers
  - function overloading
  - function pointers
  - functions with default parameters
  - pointers
  - variadic functions

  You can freely choose a feature that is not in the examples, but in that case, we recommend you to email us to check whether your choice is acceptable.
- You must describe the feature in the essay in your own words.
- You must include code examples using the feature in the essay.
- You must give links to online materials about the feature in the essay. We recommend you to find a specific section explaining the feature from the langauge specification. However, you can refer to any kinds of materials: books, blog articles, and so on.

2.3 Growing a Language

- Choose a language, such as `FAE`, from the course.
- Think how the feature can be beneficial to the language.
- You must justify the benefits in the essay. First, explain limitations of the original language. Then, show that the new feature can overcome the limitations with code examples.
- Extend the abstract syntax of the language to include the feature.
- You must define the abstract syntax in the essay. You may use the Backus–Naur form.
- Extend the operational semantics of the language to support the feature.
- You must define the big-step operational semantics with inference rules and explain them in the essay.
- Implement an interpreter of the language with the new feature. You should follow the instructions below:

  (1) Place the `midterm` directory under the `src/main/scala/cs320` directory of your SBT project for Exercise #1. The `midterm` directory contains four files:
  `honorcode.docx`, `midterm.pdf`, `Midterm.scala`, and `package.scala`.
  (2) Implement the abstract syntax in the `Midterm` trait of the `Midterm.scala` file. You do not need to implement a parser.
  (3) Write test cases for your interpreter in the `tests` function of the `package.scala` file.
  (4) Implement the `interp` (or any name you prefer) function in the `midterm` object of the `package.scala` file.
  (5) Run the tests and debug until all the tests succeed.

  The following is an example:

  – `Midterm.scala`

```scala
package cs320

trait Midterm extends Homework {
  sealed trait Expr
  case class Num(n: Int) extends Expr

  sealed trait Value
  case class NumV(n: Int) extends Value

  type Env = Map[String, Value]
}
```

  – `package.scala`

```scala
package cs320

package object midterm extends Midterm {
  def interp(expr: Expr, env: Env): Value = expr match {
    case Num(n) => NumV(n)
  }

  def tests: Unit = {
    test(interp(Num(1), Map.empty), NumV(1))
  }
}
```

3 ESSAY FORMAT

- A4, 1 inch margins at the top and bottom and on both sides
- Times New Roman, 12 points, double spacing
- Any monospaced typefaces for code snippets
- Any typefaces for syntax and semantics (you may use pictures of your handwriting)
- Maximum 6 pages

4 GRADING

You can get maximum 25 points.

4.1 Watching a Video (5 pts.)

- Summary of the content (2 pts.)
- Opinions on the video (3 pts.)

4.2 Choosing a Feature (5 pts.)

If we think that your feature is unacceptable, you will get zero points out of five points. Therefore, if you are not sure whether your choice is proper, please talk to us. Mind that you should choose a feature from a real-world language.

- Description of the feature (2 pts.)
- Code examples (2 pts.)
- Links to materials (1 pt.)

4.3 Growing a Language (15 pts.)

- Justification of benefits (5 pts.)
- Abstract syntax and operational semantics (5 pts.)
- Implementation (5 pts.)

If the interpreter does not run, then you will get zero points out of five points for implementation.

4.4 Bonus Points (5 pts.)

To encourage you to challenge difficult features, we will give bonus points of maximum 5 points for your choice of a new language feature. If you choose one of the examples that we have provided in Section 2.2, you cannot get bonus points, but you can still achieve full marks by completing everything perfectly. If you choose a feature that is not one of the examples, you can get zero or more bouns points.

5 SUBMISSION

You must submit your answer by 23:59:59 UTC+09:00, May 15th. We disallow late submissions. If you submit multiple times, only the last submission will be used for grading. We expect that your submission consists of a single zip file that contains `Midterm.scala`, `package.scala`, `honorcode.pdf`, and `essay.pdf`. You can submit the file on `kaist-cs320.appspot.com`. If you cannot reach the website, email to `cs320_ta@kaist.ac.kr` so that we can handle your submission individually.