

Darrel Chang
Lan Yang
CD 4080
9/16/24

Homework 1

Chapter 1 & 2

1. What arguments can you make against the idea of a single programming language for all domains?

Using only one single programming language for everything would not be a good idea because they each have their own strengths and weaknesses. They are made specifically for specific domains in order to excel in that category of work.

2. Name a domain specific programming language, describe its targeted usage, and explain the difference between this language and general purpose languages.

SQL is a programming language specifically used for databases. SQL has database operations that are not present in other languages such as creating, reading, updating, and deleting.

3. Describe some design trade-offs between efficiency and safety in some language you know.

In Java, the memory is automatically managed using a garbage collector that keeps track of the number of references to a variable, or object. This improves memory safety, but reduces the efficiency because it is always running and monitoring.

4. Give a brief general description of a mark-up programming hybrid language.

HTML standing for hyper text markup language is used for creating the content of webpages, using different tags to represent different kinds of content.

Chapter 3

5. Syntax of programming languages (4 points)

- (a) Use your own words to define the “syntax” of programming languages.

The syntax of a programming language defines the rules for valid code. This is in the form of how things are defined and used in a language.

- (b) Name a formal method (other than CFG) that is used for describing syntax.

Backus Naur Form (BNF) is also used for describing syntax. In BNF form, production rules are defined for what constitutes are possible.

- (c)

Write 1 or 2 lines of Java code with at least two (different) syntax errors. Describe the “nature” (i.e. what kind) of each syntax error

1: `System.out.print("Hello World!")`

The error is a lack of semicolon

2: `x = 5`

Here there is no type assigned to the variable.

6. Semantics of programming languages (4 points)

(a) Use your own words to define the “semantics” of programming languages.

The semantics of a programming language relates to the behavior of what code will do.

(b) Name one popular method used for describing semantics.

A popular method used for describing semantics is denotation semantics, which assigns mathematical functions to pieces of code and maps them to corresponding behaviors.

(c) Write 1 or 2 lines of Java code with at least two (different) semantic errors. Describe the “nature” of each semantic error (i.e. what kind of semantic error or why it is considered as semantic error.)

```
int[5] array = new int[5];  
x = array[6]
```

This is an array index out of bounds error

```
int x = 1/0;
```

divison by zero error

7. Write EBNF descriptions to write syntax rules for Java’s while loop statement. (2 points)

Notes:

(1) must use $\langle \rangle$ for an entity to be defined, i.e. $\langle \text{while_statement} \rangle$ object

(2) use $:=$ or \rightarrow to separate left side of definition and right side of definition, e.g.

$\langle \text{while_statement} \rangle$

\rightarrow ... right side of definition ...

(3) tokens/terminals must be placed in ‘ ‘, i.e. ‘while’, ‘(’, etc.

(4) you don’t have to further define expressions, other statements, etc.

$\langle \text{while_statement} \rangle := \text{'while' ' (' } \langle \text{expression} \rangle \text{')' } \langle \text{statement} \rangle$

8. Use virtual machine code (i.e. instruction like pseudo code specified in Chapter 3.5 of text, do not use MIPS instructions) to describe the operational semantics of the following C++ or Java like statements. Note: translate the following code to virtual machine code directly without any optimization. (2 points)

//assume variable j and total are properly declared and initialized

```
if (j % 2 == 0)
```

```
total += j * 2;
```

```
else
```

```
total *= j+2;
```

```
total += 1
```

Notes: (1) if-else is not a proper virtual machine code. Should use if ... goto ...

(2) $x = y * 2 + z$ isn’t a proper virtual machine code. Should use: $t1 = y * 2$, $x = t + z$ (here

assume t1 is a temporary variable)

(3) make sure each action is very specific that could match with an assembly/machine instruction.

// Assume total and j are variables in memory. Using t1, t2 registers

t1 = j % 0

if t1 == 0 goto ElseBlock

t2 = j * 2

total = total + t2

goto EndIf

ElseBlock:

t2 = j + 2

total = total + t2

EndIf:

total = total + 1