Principles of Programming Languages 202

Assignment 1

Ben Gindi - 205874142
Shira Segev - 208825349

Part 1: Theoretical Questions

Question no.1- dimensions of variability across programming paradigms

Answer:

1. Performance: how code can be run fast, use less resources (RAM, disk, network), behave better (responsive, scalable) at runtime.
2. Syntax: how natural, brief, readable is the expression of code given the syntax of the language. Can the syntax of language be extended by the programmer.
3. Domain: to which application domain is the paradigm best applicable (server-side processing, database, GUI front-end, control system).

Note: In the first lecture we learned 4 more dimensions of variability across programming paradigms, which are of course equally important.

Question no.2- Functions Types

Answer:

1. **(x , y) => x + y**

```
(x : number, y: number) : number => x + y
```

2. **x => x[0]**

```
<T>(x : T[]) : T => x[0]
```

3. **(x, y) => x ? y : -y**

```
(x : Boolean, y : number) : number => x ? y : -y
```

Question no.3- shortcut semantics

Answer:

In order to define what “shortcut semantics” are, we will use 3 additional definitions:

- Some (pred)- is a method that returns TRUE if and only if *at least one* element in an array satisfies a given predicate.
- Every (pred)- is a method that returns true if and only if *all* the elements in an array satisfy a given predicate
- Programming Functions Equivalence- 2 programming functions f and g are equivalent if they have the same domain, same range and for all values in the domain:
    - Either $f(x)$ has a value and $g(x)$ has the same value
    - Or $f(x)$ does not terminate and $g(x)$ does not terminate as well
    - Or $f(x)$ throws an error and $g(x)$ throws an error as well

Definition & Example:

Shortcut Semantics- A principle in which the condition evaluation is terminated when a conclusion is already determined.

For instance, in JAVA there is a difference between a single & or | to a double && or ||. We use the single form in order to continue checking all items (even after we find a FALSE in case of & or a TRUE in case of |), and the double form, the shortcut semantics, in order to stop evaluation when the first term that determines the result is evaluated (FALSE in case of & or TRUE in case of |).

When it comes to TypeScript, shortcut semantics is defined using the SOME and EVERY methods.

(“The native SOME and EVERY methods employ a concept known as 'shortcut semantics'. What this means, is that SOME stops and immediately returns TRUE at the moment it finds an element that satisfies the predicate. EVERY stops and immediately returns FALSE at the moment it finds an element that does not satisfy the predicate”).

Note: It is difficult to distinguish between shortcut semantics and non-shortcut semantics because both shortcut and non-shortcut versions return the same values for all parameters.