Principles of Programming Languages 232
Assignment 1 Solution

March 20, 2023

Part 1: Theoretical Questions

1. (a) Explain the following programming paradigms:
    i. Imperative - The control flow is an explicit sequence of commands.
    ii. Procedural - Same as imperative programming but organized around hierarchies of nested procedure calls.
    iii. Functional- Computation proceeds by (nested) function calls that avoid any global state mutation and through the definition of function composition.

   (b) The procedural paradigm improves over the imperative paradigm by adding layers of abstraction in the form of procedures. Procedures interact through well-defined contracts and can encapsulate local variables.

   (c) The functional paradigm improves over the procedural paradigm by discouraging the use of shared state and mutation, which makes testing, formal verification, and concurrency easier.

2. Convert the following function to adhere to the Functional Programming paradigm, using some or all the functions we saw in class: map, filter, reduce:

```
const sumEvenFP = (numbersAsString: string[]): number =>
    numbersAsString
    .map((x: string): number => parseInt(x, 10))
    .filter((x: number): boolean => x % 2 == 0)
    .reduce((acc: number, curr: number): number => acc + curr, 0)
```

3. Write the most specific types for the following expressions:
   (a) `<T>(x: T[], y: (z: T) => boolean) => boolean`
   (b) `(x: number[]) => number`
   (c) `<T>(x: boolean, y: T[]) => T`
   (d) `<T1, T2>(f: (b: T1) => T2, g: (a: number) => T1) => (x: number) => T2`

4. Abstraction barriers isolate different “levels” of the system. The implementer of the high-level system doesn’t need to know about low-level details.