

Learning to Program with F#
Exercises
Department of Computer Science
University of Copenhagen

Jon Sparring, Martin Elsmann, Torben Mogensen, Christina Lioma

October 21, 2022

0.1 Modules

0.1.1 Teacher's guide

Emne Moduler, namespaces og afprøvning

Sværhedsgrad Middel

0.1.2 Introduction

A color is often represented as a triple (red, green, blue), where each entry is called a color-channel, and each channel is typically an integer between and including 0 and 255:

$$c = (r, g, b). \quad (1)$$

Colors can be added, by adding their channels,

$$c_1 + c_2 = (\text{trunc}(r_1 + r_2), \text{trunc}(g_1 + g_2), \text{trunc}(b_1 + b_2)), \quad (2)$$

$$c_i = (r_i, g_i, b_i), \quad (3)$$

$$\text{trunc}(v) = \begin{cases} 0, & v < 0 \\ 255, & v > 255 \\ v, & \text{ellers} \end{cases} \quad (4)$$

and colors can be scaled by a factor by multiplying each channel with that same factor,

$$ac = (\text{trunc}(ar), \text{trunc}(ag), \text{trunc}(ab)). \quad (5)$$

Colors where the channels have identical values, $v = r = g = b$, are grays, and colors are converted to grays as the average,

$$v = \text{gray}(c) = \frac{r + g + b}{3}. \quad (6)$$

0.1.3 Exercise(s)

- 0.1.3.1:** Write a signature file for a module which contains the functions `trunc`, `add`, `scale`, and `gray` from the mathematical definitions above and use tuples where possible.
- 0.1.3.2:** Write an implementation of the signature file from Assignment 1 and compile both files into a library (dll-file).
- 0.1.3.3:** Write two programs: One which uses the library developed in Assignment 1 and 2 using `fsharp_i` and one which uses `fsharp_c`.
- 0.1.3.4:** Make a Black-box test of your library from Assignment 2.
- 0.1.3.5:** Make a White-box test of your library from Assignment 2.

0.1.3.6: Consider the library from Assignment 2. Assuming that your module is called `Color`, consider the following application

Listing 1: Application of a Color library.

```
1 let red = (255,0,0)
2 let green = (0,255,0)
3 let avg = Color.add red green
4 let factor = 1.25
5 let bright = Color.scale factor avg
6 printfn "Bright gray is: %A" bright
7
```

If your functions `add` and `scale` have a different interface, then adjust accordingly. Perform a tracing by hand of the above code including the implementation of your library. Run the (adjusted) code with `fsharpc`. Did you discover any errors? Do you get the same output?

0.1.3.7: Extend the library (both the signature and the implementation file) from Assignment 1 and 2) with a function that converts a color tripple into a tripple of identical gray values. Extend your test with a suitable set of tests of this new function. Discuss whether the library, application, and test are structured in a way such that the extension has been easy, or whether there are dependencies that makes correcting, maintaining, extending the code difficult and with a high risk of introducing new errors.