

exercise09-p

December 14, 2018

0.0.1 Assignment 9.1 (P) (Delayed) evaluation, unit, side-effects, pure functions

We can see let-bindings as functions without any arguments. Therefore they can and will be immediately evaluated. If we want to delay evaluation of the bound expression until application, we can introduce an argument. However, if there are no free variables in our expression, we just need some dummy argument. To indicate this, we use the type unit which has only one value () (which can be seen as the empty tuple).

Discuss this difference between the following two expressions

```
In [ ]: let x = print_endline "foo" in x, x
```

```
In [ ]: let x () = print_endline "foo" in x (), x ()
```

1. What are side-effects? Give some examples.
2. What are pure functions? What are their benefits?
3. Why does delaying evaluation only make sense in case of side-effects?
4. Why do we want to use () instead of some unused variable?

0.0.2 Assignment 9.2 (P) Students in, students out!

Once again, we consider our student records:

```
In [ ]: type student = {  
    first_name : string;  
    last_name  : string;  
    id         : int;  
    semester   : int;  
    grades     : (int * float) list  
}  
  
type database = student list
```

Now, we define a file format to store students that, for each student, contains a line
first_name;last_name;id;semester;gc
where gc number of lines
course;grade
follow with grades.

- 1) Implement a function `load_db : string -> database` to load the students from the given file. Throw an exception `Corrupt_database_file` if something is wrong with the file.
- 2) Implement a function `store_db : string -> database -> unit` to store the students back to the given file.