Problem Set 4
COMP301 Fall 2019
24.10.2019 17:30 - 18:45

To do these problems, first download the `let` language files from `https://github.com/racket/eopl/tree/master/tests/chapter3/let-lang`. You will be modifying some of the code in there. When you are done, you can test the `minus` operation in *tests.rkt* from the console and you can test `extend-env*` in *environments.rkt* from the console. We will also be using EOPL GUI, which you can find under Course Content on Blackboard. In this problem session, you can try modifying the let language in the EOPL GUI under the languages folder, then in the GUI you can test the `minus` operator.

Problem 1[1]: Extend the `Let` language by adding a new operator `minus` that takes one argument `n` and returns `-n`. For example, the value of `minus(-(minus(5),9))` should be `14`, which is equal to $-((-5)-9)$.

Problem 2: Write an `extend-env*` function, which works identical to `extend-env` but can take many parameters. Below is an example:

```
> (extend-env
   `i (num-val 1)
   (extend-env
      `v (num-val 5)
      (extend-env
        `x (num-val 10)
        (empty-env))))
((i #(struct:num-val 1)) (v #(struct:num-val 5)) (x #(struct:num-val 10)))

> (extend-env* `(x v i) (list (num-val 10) (num-val 5) (num-val 1))
   (empty-env))
((i #(struct:num-val 1)) (v #(struct:num-val 5)) (x #(struct:num-val 10)))
```

You can of course use the pre-existing `extend-env` in your function.

[1] EOPL p.72 Exercise 3.6