Problem Set 7
COMP301 Fall 2019
28.11.2019 17:30 - 18:45

Read me first! Please download the *Codes* file. In the scheme codes, you will see some hints regarding where to modify. You will use DrRacket. We have also edited the `tests.rkt` so that if you solve the problem, running `tests.rkt` should have no errors. You have only one language to modify in this PS, and both questions are very similar in terms of modification.

Problem 1[1]: Add arrays to `mutable-pairs` language. Introduce new operators `newarray`, `arrayref`, and `arrayset` that create, dereference, and update arrays. This leads to:

$$ArrVal = (Ref(ExpVal))^*$$
$$ExpVal = Int + Bool + Proc + ArrVal$$
$$DenVal = Ref(ExpVal)$$

Since the locations in an array are consecutive, use a representation like the second representation above. What should be the result of the following program?

```
let a = newarray(2, -99)
   p = proc (x)
      let v = arrayref(x,1)
      in arrayset(x,1,-(v,-1))
in begin
   arrayset(a,1,0);
   (p a);
   (p a);
   arrayref(a,1) end
```

Here `newarray(2,-99)` is intended to build an array of size 2, with each location in the array containing -99. `begin` expressions are defined already for you (see exercise 4.4 for them). Make the array indexing zero-based, so for example an array of size 2 should have indices 0 and 1.

Problem 2[2]: Add to the language of exercise 4.29 (previous problem) a procedure `arraylength`, which returns the size of an array. Your procedure should work in constant time. Make sure that `arrayref` and `arrayset` checks that their indices are within the length of the array.

[1] EOPL p.128-129, Exercise 4.29
[2] EOPL p.130, Exercise 4.30