Problem Set 9
Comp 301
Fall 2022
Week 11: 19.12.2022 - 23.12.2022

Instructions:

- Submit your answers to the Blackboard PS8 assignment until December 24th Saturday, at 23.59.
- Please use the code boilerplate, which includes several tests for you to see if your code is correct.
- Save your code and pdf file, zip it as `ID_username.zip` with your ID and username (Example: `1234567_oarpaci18.zip`), and submit this ZIP file.
- Read the questions carefully. Good luck!

Problem 1: [1] In `IREF`, why are variables in environment bound to integers rather than expressed values, as in `EREF`?

Problem 2: [2] Modify the implementation of `proc-exp` to allow multiple arguments. Also change the implementation of `call-exp` to accept multiple operands.

Hint: You have implemented multi-argument procs before. Now try to do it in IREF.

New grammar of `proc-exp` and `call-exp`

$$\begin{aligned} \mathit{Expression} &:= \text{proc}(\{\mathit{Identifier}\}^{*(,)})\ \mathit{Expression} \\ &:= (\mathit{Expression}\ \{\mathit{Expression}\}^{*}) \end{aligned}$$

Example

```
let x = 5 in let y = 6 in
   let z = proc (a,b) -(a,b) in (z x y)
=> returns (num-val -1)
```

Recommended Changes:

- **Note:** Carefully read the instructions commented in the code as well. More detailed information is given in the comments.
- Change `proc` datatype and implement `extend-env*` ,which extends environment with multiple var val pairs at once, in `data-structures.rkt`
- Change `apply-env` to handle the `extend-env*` in environments.rkt
- Change the implementations of grammars of `proc-exp` and `call-exp` accordingly in `lang.rkt`
- Change the implementation of `proc-exp`, `call-exp` inside value-of and also change apply-procedure to evaluate multi-argument procedures in `interp.rkt`.

Problem 3: [3] Modify the let expression so that it extends the environment with multiple pairs at once.

New grammar of `let-exp`

$$\mathit{Expression} := \text{let}\ \{\mathit{Identifier} = \mathit{Expression}\}^{*(,)}\ \text{in}\ \mathit{Expression}$$

[1] EOPL p.121 Exercise 4.15

[2] EOPL Exercise 4.17

[3] EOPL Exercise 4.17

Example

```
let x = 5, y = 6 in -(x,y)
=> returns (num-val -1)
```

Recommended Changes:

- **Note:** Carefully read the instructions commented in the code as well. More detailed information is given in the comments.
- Change the implementation of grammar of let-exp accordingly in `lang.rkt`
- Change the implementation of let-exp in `interp.rkt`
- **Hint**: You can use previously implemented `extend-env*`