

MAC0316/5754

Exercício Programa 1

Data de entrega: 09/11/2020, 8:00hs

Instruções:

1. Você deve entregar seu programa pelo PACA em um **único arquivo** *.rkt* contendo as definições do interpretador
2. Programas **atrasados** terão uma **penalidade de 20%** na pontuação **POR DIA**.

Interpretador (10 pontos)

- a) (4 pontos) Implemente `let*` e `letrec` como visto em classe. Suporte um número fixo de argumentos (2 e 1, respectivamente). Você deve implementar `let*` como função “core” do seu interpretador (ou seja, deve haver uma `ExprC` equivalente) e `letrec` como açúcar sintático (ou seja deve haver apenas uma `ExprS` associada).
- b) (2 pontos) Implemente a primitiva `quote`, que gera um símbolo, com `'` no seu interpretador. Assim:

```
(interpS `(quote alan))
```

deve retornar

```
- symbol  
(symV 'alan)
```

- c) (4 pontos) Implemente a primitiva `read`, sem argumentos, que abre uma janela de leitura e lê uma expressão simbólica para ser avaliada. Assim:

```
(interpS (load))
```

deve abrir uma janela para o usuário, onde este deve digitar o programa a ser avaliado. Para isso use a primitiva “read” da linguagem `plai-typed`. Note que na janela que será aberta, você pode colar uma expressão com mais de uma linha como:

```
(seq (+ 2 3)  
      (- 4 5)); na nossa linguagem “begin” é “seq”
```

IMPORTANTE: Não será dado **crédito parcial** aos itens acima. Apenas aqueles que funcionarem receberão pontos (e.g. se você implementar `let` no parser mas não no interpretador, a implementação do `let` está incompleta).

IMPORTANTE: Você deve usar a versão `plai-typed` do interpretador com fechamentos *`closureTyped.rkt`* (disponibilizado com este enunciado). Assim, você poderá utilizar os comandos de gerenciamento de arquivos do Scheme. (**Dica:** procure a documentação do comando `read`).