

# Avaliação prática: implementação de um verificador de *frames* para programas PDL

Linguagens de Programação

## 1 Definição

Implementar um programa em uma linguagem de programação funcional que receba como entrada um *frame* PDL (i.e. um grafo com arestas rotuladas)  $\mathcal{F} = \langle W, R_\alpha \rangle$  tal que  $W$  é um conjunto de mundos (vértices) e  $R_\alpha$  é uma relação binária indexada (arestas rotuladas) e um programa PDL  $\pi$ . A saída deve ser o resultado da verificação de se  $\mathcal{F}$  corresponde a um *frame* válido para  $\pi$ , isto é, se  $\mathcal{F}$  é um grafo induzido por  $\pi$ . Caso a resposta seja negativa, uma mensagem de erro deve explicitar o ponto de incompatibilidade.

## 2 Da implementação

O programa deve ser feito em alguma linguagem de programação funcional como Haskell, Racket ou Scheme, fazendo uso apenas de instruções típicas de linguagens funcionais. Mesmo que a linguagem dê suporte a instruções de paradigma imperativo e/ou outros paradigmas, essas não poderão ser utilizadas. Demais linguagens de programação devem ser consultadas previamente. Demais ferramentas devem ser consultadas previamente.

## 3 Entregáveis

A entrega do trabalho constará de

- i) código fonte (com Makefile e/ou roteiro completo de compilação/execução) do programa,
- ii) arquivos de exemplo e
- iii) apresentação do trabalho.

## 4 Prazo

**Entrega por email:** até o dia 12 de novembro de 2019

**Apresentação:** até a aula do dia 13 de novembro de 2019