

1 Instruções Importantes

Nessa seção são apresentadas diversas informações relevantes referentes a entrega do trabalho e orientações a serem seguidas durante a implementação do mesmo. Leia atentamente antes de começá-lo.

1.1 Equipe de Desenvolvimento

O trabalho será desenvolvido individualmente ou em dupla.

1.2 Linguagem de Programação

O trabalho deverá ser desenvolvido na linguagem Racket.

1.3 Artefatos a Serem Entregues

Os artefatos a serem entregues são:

- código fonte do interpretador contido no arquivo *interpreter.rkt* e módulos adicionais que venham a ser criados;
- relatório do trabalho em formato pdf.

Antes de enviar seu trabalho para avaliação, assegure-se que:

1. seu código executa no REPL de Racket. Programas com erros de sintaxe receberão nota zero;
2. todos os fontes a serem enviados têm, em comentário no início do arquivo, nome e matrícula do autor do trabalho;
3. arquivo de relatório tenha a identificação do autor do trabalho;

1.4 Critérios de Avaliação

A avaliação será feita mediante análise do código fonte, relatório e apresentação do trabalho (entrevista). Os seguintes fatores serão observados na avaliação do código fonte: correteza do programa, estrutura do código, redigibilidade e legibilidade. A correteza se refere à implementação correta de todas as funcionalidades especificadas, i.e., se o programa desenvolvido está funcionando corretamente e não apresenta erros. Os demais fatores avaliados no código fonte são referentes a organização e escrita do trabalho.

O relatório do trabalho deverá relatar as decisões de projeto e estratégia intelectual para solução do trabalho. Espera-se que no relatório tenha informações a respeito das estruturas de dados utilizadas, representação dos valores das expressões (*expressed values*), valores denotáveis (*denoted values*) e uma descrição em alto nível da solução. Recomenda-se a utilização de exemplos para ilustrar e/ou detalhar a abordagem utilizada para resolver o problema.

O trabalho deverá ser apresentado ao professor da disciplina e, só será avaliado após a realização da entrevista, i.e., trabalhos que não forem apresentados não terão nota. Na entrevista, o discente deverá elucidar, ao menos, como modelou e resolveu o problema. A entrevista também tem a finalidade de avaliar a confiabilidade e segurança do autor do código em explicar pontos relevantes do trabalho desenvolvido.

Assim, a entrevista influenciará na avaliação dos artefatos entregues. Portanto, a nota final será dada a partir da avaliação do conjunto do código fonte, relatório e entrevista. **É de responsabilidade do discente solicitar a marcação do dia e horário da entrevista com o professor da disciplina.**

Dias de Atraso	Nota
1	$n * 0.98$
2	$n * 0.96$
3	$n * 0.92$
4	$n * 0.84$
5	$n * 0.68$
6	$n * 0.36$
7	0

Program	→	{Clause}*
Clause	→	Head .
		Head :- {Body}* ^(,) .
Body	→	Term
		Term = Term
		Term \= Term
Term	→	VAR
		Head
Head	→	ATOM
		Functor
Functor	→	ATOM ({Arg}* ^(,))
Arg	→	Term
		-

Figura 1: Sintaxe da linguagem ICLASSES

Atrasos serão penalizados por uma função exponencial de dias de atraso, i.e., será reduzido da nota um percentual referente a exponencial na base 2 dos dias de atraso. A tabela a seguir mostra a nota em função dos dias de atraso:

Observe que a partir do 7º dia de atraso seu trabalho não será mais avaliado.

2 A Linguagem LOGIC

LOGIC é uma linguagem lógica cuja sintaxe é inspirada no prolog. Assim como em Prolog, um programa em LOGIC consiste em uma base de conhecimento formada por fatos e regras, que descrevem relações entre objetos e conceitos. A execução do programa ocorre por meio de consultas, em que o interpretador tenta inferir a resposta com base nos fatos e regras definidos.

A gramática da Figura 1 especifica a sintaxe concreta da linguagem, no formato EBNF, destacando em negrito os símbolos terminais. Ressalta-se que a notação $\{A\}^*$ denota uma sequência do elemento A e a notação $\{A\}^{*(,)}$ denota uma sequência de elementos A separados por vírgula.

Os terminais **ATOM** e **VAR** representam constantes e variáveis, respectivamente, e seguem a mesma definição de Prolog. As constantes são formadas por uma sequência de símbolo que começam com uma letra minúscula e as variáveis começam com a letra maiúscula ou com *underline*. A variável *-* (*underline*) denota uma variável anônima. Comentários de linha são definidos usando o símbolo % (porcento).

Um exemplo de programa na linguagem é apresentado na Figura 2. As consultas usam a mesma sintaxe do corpo de uma cláusula.

```
parent(pam, bob). % pam é um dos pais de bob

parent(tom, bob). % tom é um dos pais de bob
parent(tom, liz). % tom é um dos pais de liz

parent(bob, ann). % bob é um dos pais de ann
parent(bob, pat). % bob é um dos pais de pat

parent(pat, jim). % pat é um dos pais de jim.

granparent(X,Y) :- parent(X,Z), parent(Z,Y).
```

Figura 2: Exemplo de programa na linguagem ICLASSES

3 Especificação Técnica do Trabalho

Com base no conteúdo estudado na disciplina, neste trabalho pede-se para implementar a linguagem LOGIC usando Racket. Para realizar a implementação deste trabalho, você deverá usar o código disponível no github da disciplina ¹. O código disponibilizado traduz a sintaxe concreta da linguagem para uma representação abstrata (*AST*), descrita no arquivo *ast.rkt*. Esta *AST* será a interface que deverá ser usada para realizar a implementação do interpretador para a linguagem.

A função *eval-query* no arquivo *interpreter.rkt* é o ponto de entrada para uma consulta. Após carregado o programa, o usuário poderá realizar consultas no REPL do racket. A consulta do usuário e o programa são parâmetros da função *eval-query*, os quais devem ser usados para processar a consulta do usuário e emitir uma resposta.

Assim, as tarefas que devem ser realizadas por você são:

- construir uma representação adequada para o programa que facilite o mecanismo de inferência;
- usar a resolução SLD para realizar a consulta do usuário.

Na pasta *examples* há um conjunto de programas que podem ser utilizados para testar a sua implementação. O único arquivo que deve, e pode, ser alterado é o *interpreter.rkt*. Os demais arquivos não devem sofrer alterações e não precisam ser entregues. Caso julgar necessário, você pode criar outros arquivos.

4 Entrega do Trabalho

A data da entrega do trabalho será até o dia **09 de março de 2025**, via plataforma do GoogleClassroom. A entrevista deverá ser realizada juntamente com o professor da disciplina até o dia **20 de março de 2025**.

¹<https://github.com/lvsreis/dcc019/exercise/logic>