

**Lista de Exercícios de Linguagens de Programação III**  
**Universidade Federal do Amazonas / Instituto de Computação**  
**Marco Cristo**

**Teoria**

- 1) Entre as características associadas com linguagens lógicas, podemos citar a programação baseada em motores de inferência, a unificação, a representação via cláusulas lógicas e a falsidade como falha. O que você entende por cada uma delas?
- 2) Por que o uso de predicados *assert* e *retract* é associado à programação com efeitos colaterais?

**Programação em Lógica**

- 3) Traduza as seguintes sentenças (em *itálico*) para Prolog. Consulte então o programa para responder as perguntas que seguem o texto.

**Texto:** *Chiara foi assassinada com um instrumento semelhante a um bastão. A morte ocorreu provavelmente na quinta ou sexta-feira. Nada indica que mais que uma pessoa esteve envolvida. As evidências apontam que o motivo do crime foi dinheiro, ciúmes ou insanidade, e que o assassino deve ser uma das pessoas que moravam na pensão com Chiara. Os investigadores sabem que a arma do crime ou é a perna de madeira que foi roubada de Pietro (um rapaz pobre) na quinta-feira em Belo Horizonte ou na quarta-feira em Contagem; ou o martelo que foi roubado da caixa de ferramentas da pensão na quarta ou quinta-feira. O assassino entrou no quarto de Chiara usando uma chave que foi roubada dela na segunda-feira em Contagem ou terça-feira em Belo Horizonte. Uma quantia considerável de dinheiro foi levada do quarto da Chiara. A amiga pobre de Chiara, Eliza, tem uma cópia da chave. Chiara tem um relacionamento com Pietro, que por sua vez teve um relacionamento com uma garota rica, a Carol. Além disso, Chiara tinha um relacionamento com Alfredo, que é pobre, mas se relacionava com uma garota rica, Elena, que por sua vez tinha um relacionamento com o igualmente rico Henrico. Henrico estava romanticamente envolvido com Maria, uma garota pobre, que costumava se envolver com Adriano, que é rico, e costumava se envolver com Carol. Alfredo estava em Contagem na segunda e terça-feira, em Belo Horizonte na quarta-feira, em Contagem novamente na quinta-feira, quando voltou à pensão na sexta. Carol esteve em Belo Horizonte de segunda a quarta-feira, em Contagem na quinta-feira e na sexta-feira voltou à pensão. Henrico estava na pensão na segunda-feira, em Belo Horizonte, na terça-feira, e depois voltou para a pensão. Eliza passou a segunda-feira na pensão, esteve em Belo Horizonte na terça e quarta-feira, em Contagem na quinta-feira, e voltou para casa na sexta-feira de manhã. Adriano estava em Contagem na quarta-feira e em casa pelo resto da semana. Elena estava em Belo Horizonte terça e quarta-feira, na pensão segunda, quinta e sexta-feira. Pietro esteve em Contagem segunda e terça-feira, em Belo Horizonte na quarta-feira, em Contagem*

novamente na quinta-feira, e em casa sexta-feira. Maria estava em Contagem de terça a quinta-feira e na pensão de segunda a sexta-feira. Adriano e Maria têm consultado um terapeuta por causa de distúrbios psicóticos que ocasionalmente se manifestam como comportamento violento sem motivação. **Pergunta:** Quem matou Chiara e por quê?

4) Implemente um interpretador em Prolog que traduza sentenças em linguagem natural para o protocolo reconhecido pelo servidor Python descrito na **Lista 2**. Ou seja, este cliente deve receber sentenças em linguagem natural e emití-las em um dos formatos abaixo:

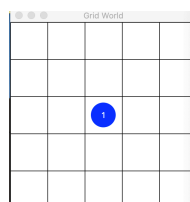
- + id F C X Y: criar figura com identificador id (inteiro), forma F (s para quadrado e c para círculo), cor C (pelo menos, *black*, *red*, *green* e *blue*) e coordenadas X e Y (inteiros de 0 a 4).
- - id: apagar figura identificada por id.
- - F C: apagar figura identificada por F C.
- m id X Y: mover figura identificada por id para coordenadas X e Y.
- m F C X Y: mover figura identificada por F C para coordenadas X e Y.

A seguir, temos exemplos de sentenças e suas traduções, correspondentes a exemplos dados na **Lista 2**.

Sentença entrada pelo usuário	Sentença interpretada
desenhar circulo azul em 2 2	+ 1 c blue 2 2
criar quadrado vermelho na posicao 1 3	+ 2 s red 1 3
mover circulo azul para 1 2	m c blue 1 2
mover quadrado vermelho para esquerda do circulo azul	m s red 1 1
apagar quadrado vermelho	- s red
deslocar quadrado vermelho 2 2	m s red 2 2
inserir quadrado verde abaixo de circulo azul	+ 3 s green 2 2
mover 3 para baixo do 1	m s green 3 2
apagar 1	- c blue

Note que as sentenças envolvem identificadores (1, c blue), sinônimos (desenhar ~ criar ~ inserir, etc), instruções relativas (para esquerda do, abaixo do, acima do, a direita do, etc) e algumas simplificações (deslocar quadrado vermelho 2 2). A tradução de instruções relativas pressupõe que seu programa seja capaz de lembrar das posições de figuras previamente colocadas no grid. A seguir temos exemplos da execução esperada do cliente Prolog, quando se comunicando com o meu servidor da **Lista 2**.

```
?- consult('interpret.pl').
true.
```



```

?- go_socket.
>> criar circulo azul em 2 2
Enviado: + 1 c blue 2 2
Respost: Done.

>> apagar 1
Enviado: - c blue
Respost: Done.

>> criar circulo azul em 2 2
Enviado: + 2 c blue 2 2
Respost: Done.

>> criar quadrado vermelho aa esquerda do circulo azul
Enviado: + 3 s red 2 1
Respost: Done.

>> mover 3 para 1 1
Enviado: m s red 1 1
Respost: Done.

>> criar quadrado preto aa esquerda do 2
Enviado: + 4 s black 2 1
Respost: Done.

>> criar quadrado verde acima do 2
Enviado: + 5 s green 1 2
Respost: Done.

>> mover 3 abaixo dos 2
Nao entendi o que vc disse. Pode repetir?

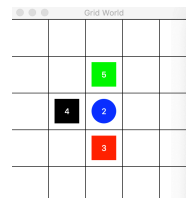
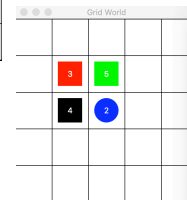
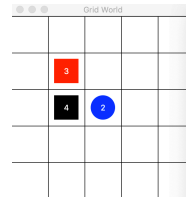
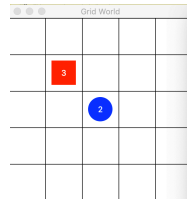
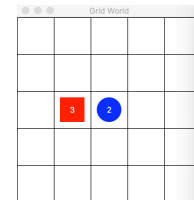
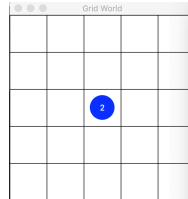
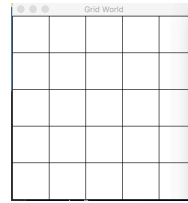
>> mover 3 para baixo do 2
Enviado: m s red 3 2
Respost: Done.

>> apagar circulo verde
Enviado: - c green
Respost: Figure not found: c green

>> mover 17 para baixo do 2
Nao entendi o que vc disse. Pode repetir?

>> bye
Tchau!
true .

```



Como ajuda inicial, é fornecido um programa Prolog capaz de traduzir a sentença ‘criar circulo azul em 1 1’ para ‘+ 1 c blue 1 1’. Este código inclui interfaces para teclado e para comunicação via socket com o servidor Python da **Lista 2**. Ele deve ser estendido para que sejam fornecidas as funcionalidades descritas na tabela dada.

```
:- use_module(library(streampool)).
```

```

/* interpretador */

:- dynamic(ultimoId/1).

/* comandos interpretados */

proximoId(Id) :-
    ultimoId(Ultimo),
    Id is Ultimo + 1,
    retractall(ultimoId(_)),
    assert(ultimoId(Id)).

criar(Forma, Cor, X, Y, Str) :-
    proximoId(Id),
    atom_number(AId, Id),
    atomic_list_concat([+, AId, Forma, Cor, X, Y], ' ', Atom),
    atom_string(Atom, Str).

/* NLP */

comando_criar(Forma, Cor, X, Y) -->
    verbo_criar, forma(Forma), cor(Cor), prep, num(X), num(Y).

prep --> [em].

forma(c) --> [circulo].

cor(blue) --> [azul].

num(1) --> ['1'].

interprete(L, Str, continuar) :-
    comando_criar(Forma, Cor, X, Y, L, []),
    criar(Forma, Cor, X, Y, Str).

interprete([bye], '', parar).

interprete(_, 'Nao entendi o que vc disse. Pode repetir?', imcompreensivel).

estado_inicial :-
    retractall(ultimoId(_)),
    assert(ultimoId(0)).

/* interface via teclado */

prompt(L) :-
    write('>> '),
    read_line_to_codes(user_input, Cs),
    atom_codes(A, Cs),
    atomic_list_concat(L, ' ', A).

interaja(parar) :-
    write('Tchau!'), nl.
interaja(_) :-
    prompt(L),

```

```

    interprete(L, Str, ProxCmd),
    write(Str), nl,
    interaja(ProxCmd).

go :-
    estado_inicial,
    interaja(continuar).

/* interface via socket */

create_client(Host, Port) :-
    setup_call_catcher_cleanup(tcp_socket(Socket),
                               tcp_connect(Socket, Host:Port),
                               exception(_),
                               tcp_close_socket(Socket)),
    setup_call_cleanup(tcp_open_socket(Socket, In, Out),
                       chat_to_server(In, Out),
                       close_connection(In, Out)).

chat_to_server(In, Out) :-
    prompt(L),
    interprete(L, Str, ProxCmd),
    (   ProxCmd == parar
    -> interaja(parar)
    ;   (   ProxCmd == imcompreensivel
    -> write(' '), write(Str), nl, nl,
        chat_to_server(In, Out)
    ;   write(' Enviado: '), write(Str), nl,
        format(Out, '~s', Str),
        flush_output(Out),
        read_string(In, '\n', '\r', _, Reply),
        write(' Respost: '), write(Reply), nl, nl,
        chat_to_server(In, Out)
    )
    ).

close_connection(In, Out) :-
    close(In, [force(true)]),
    close(Out, [force(true)]).

go_socket :-
    estado_inicial,
    create_client('localhost', 7777).

```

- 5) Baseado em sua experiência com Prolog, a compare com Python OO, considerando os seguintes critérios:
- Simplicidade, Ortogonalidade, Tipos de Dados, Projeto de Sintaxe, Suporte à Abstração, Expressividade, Checagem de Tipos, Manipulação de Exceções e Restrição de *Aliases*
  - Legibilidade, escrita e confiabilidade

### **Observações**

- Resposta pode ser entregue em dupla;
- Plágio não será tolerado, com anulação dos casos observados;
- Data de entrega a ser definida no site da disciplina;
- Além da lista, devem ser enviados os códigos fontes necessários para testar o programa implementado, junto com instruções de como testá-lo;
- A parte prática é a de maior peso nesta lista;