

Universidade de Brasília - FGA

Paradigmas de Programação

Professora: Milene Serrano

Alunos: Cleiton Gomes; Vanessa Barbosa

Modularização robô Velc – Paradigma Lógico

Brasília, maio de 2014.

Considerações iniciais

Este documento possui o intuito de evidenciar a modularização do sistema em prolog do robô Velc e fornecer uma breve explicação de cada módulo.

bancoDeDados.pl

Possui a responsabilidade de carregar os dados dos cálculos das correlações de Pearson realizados pelo programa em C e converte em uma estrutura que a linguagem Prolog entenda.

```
carregaBase:-open
('calculoPearsonEmC.txt',read,Arquivo),
read(Arquivo,Correlacao21Candles),
read(Arquivo,Correlacao34Candles),
read(Arquivo,Correlacao55Candles),
read(Arquivo,Correlacao89Candles),
close(Arquivo), write
(Correlacao21Candles),write(-), write
(Correlacao34Candles),write(-), write
(Correlacao55Candles),write(-), write
(Correlacao89Candles),write(-), tell
('tabelaDadosPearson.pl'), write
('pearson(21,')', write
(Correlacao21Candles), write(').'),nl,
write('pearson(34,')', write
(Correlacao34Candles), write(').'),nl,
write('pearson(55,')', write
(Correlacao55Candles), write(').'),nl,
write('pearson(89,')', write
(Correlacao89Candles), write(').'),nl,
nl,told.
```

controladoraFatos.pl

A regra manipulaFatosPearson possui o intuito de capturar os dados fornecidos pelo programa em C das correlações de Pearson que apresentam um bom desempenho (ganham dinheiro) e esses novos fatos são inseridos na base. Caso contrário, se os dados apresentarem desempenho inadequado, os dados são procurados na base e se existirem são excluídos através da regra excluirFato.

```

:-include('tabelaFatos.pl').manipulaFatosPearson:-
open('calculoPearsonEmC.txt',read,Arquivo),
read(Arquivo,Correlacao21Candles),      read
(Arquivo,Correlacao34Candles),  read
(Arquivo,Correlacao55Candles),  read
(Arquivo,Correlacao89Candles),  close(Arquivo),
open('saidaMql.txt',read,Arquivo2),      read
(Arquivo2,StatusCorrelacao21),  read
(Arquivo2,StatusCorrelacao34),  read
(Arquivo2,StatusCorrelacao55),  read
(Arquivo2,StatusCorrelacao89),  close(Arquivo2),
insereNovoFato(Correlacao21Candles,
Correlacao34Candles, Correlacao55Candles,
Correlacao89Candles,
StatusCorrelacao21, StatusCorrelacao34,
StatusCorrelacao55, StatusCorrelacao89),
excluiFato(StatusCorrelacao21, StatusCorrelacao34,
StatusCorrelacao55, StatusCorrelacao89).
manipulaFatosPearson.insereNovoFato
(Correlacao21Candles, Correlacao34Candles,
Correlacao55Candles, Correlacao89Candles,
StatusCorrelacao21, StatusCorrelacao34,
StatusCorrelacao55, StatusCorrelacao89):-

```

controleOperacoes.pl

Para se realizar uma ordem de compra com uma alavancagem específica, é necessário ter o controle das operações, informando ao robô em linguagem MQL, a alavancagem e informação de compra. Essas instruções são realizadas pela regra carregaCorrelacoes.

```

carregaCorrelacoes:-open
('calculoPearsonEmC.txt',read,Arquivo),
read(Arquivo,Linha1),      read
(Arquivo,Linha2),          read
(Arquivo,Linha3),          read
(Arquivo,Linha4),          alavancagem
(Linha1,Linha2,Linha3,Linha4), close
(Arquivo).alavancagem
(Linha1,Linha2,Linha3,Linha4):-      (
(Linha1 > 0.8, Cont = 0.1) , (Linha2 >
0.8, Cont = 0.2) , (Linha3 > 0.8,
Cont = 0.3 ) , (Linha4 > 0.8, Cont = 0.4)
), registraOperacao(Cont).alavancagem
(Linha1,Linha2,Linha3,Linha4).
registraOperacao(Cont):-      Cont > 0,
tell('entradaMQL.txt'), write(Cont), write
('compre'), nl,told.

```

tabelaFatos.pl

Possui a tabela binária das correlações que são significativas.

```
:- dynamic pearson/2.
pearson(21, 0.937819).
pearson(34, 0.914443).
pearson(55, 0.91677).
pearson(89, 0.93617).
```

tabelaFatosNinja.pl

Possui os fatos que podem ser levemente modificados com o intuito de robô aprender com os erros e acertos.

```
:- dynamic pearson/2.
pearson(21, 0.925787).
pearson(34, 0.944446).
pearson(55, 0.896775).
pearson(21, 0.937819).

pearson(21, mmSuperRapida).
pearson(34, mmRapida).
pearson(55, mmMedia).
pearson(21, mmRegular).
```

metodosNumericos.pl

Realiza a chamadas das regras principais e faz com que o fluxo de execução seja percorrido por toda a lógica da aplicação. É importante ressaltar que a regra main é pequena, pois as outras regras delegam responsabilidades para outras regras e assim por diante.

```
:-dynamic metodos/3.
:-include('bancoDeDados.pl').
:-include('controladoraFatos.pl').
:-include('controleOperacao.pl').

main:- carregaBase,
insereNovoFatoPearson,
carregaCorrelacoes, halt.
```