Universidade Federal de Juiz de Fora

Linguagens de Programação

Prolog – Cifra de César e cifra de Vigenère

Daniel Machado Barbosa Delgado – 201835013

1 Arquivos

Os arquivos se encontram organizados da seguinte maneira:

- main.pl : Arquivo centralizador que possibilita todas as consultas possíveis a partir de apenas um arquivo.

- code.pl : Arquivo que armazena os predicados de caracteres e seus códigos correspondentes.

- palavra.pl : Arquivo que armazena os predicados da base de palavras da língua portuguesa.

- predicados.pl : Arquivo destinado a armazenar predicados que são úteis em diversos setores do programa.

- cesar.pl : Arquivo que contem a codificação e a decodificação de textos utilizando a cifra de César.

- vigenere.pl : Arquivo que contem a codificação e a decodificação de textos utilizando a cifra de Vigenère.

2 Predicados importantes

Possuímos alguns predicados que são importantes para o bom funcionamento da aplicação, e que, acabam sendo usado mais de uma vez em contextos diferentes.

- *code(Char,Code)* : O predicado que relaciona um caractere com o seu respectivo código.

- *palavra(Palavra)* : Representa uma palavra válida da língua portuguesa.

- *string2code(String, Code)* : Importantíssimo predicado que faz uso do outro predicado code para relacionar um texto com uma lista de código de seus caracteres correspondentes. Nota-se que este predicado é usado de maneira a auxiliar tanto realização da cifra de César, quanto a cifra de Vigenère.

3 Gerenciamento de Dados.

O gerenciamento de dados foi feito a partir de alguns predicados, que ficaram responsáveis por:

- Inserir uma nova palavra no documento;

- Inserir uma nova palavra em memória com o assertz;

- Remover uma palavra em memória com o retract;

```
1 ?- insereNovaPalavraNoDocumento(testando123).
true.

2 ?- palavra(testando123).
false.

3 ?- inserePalavraNaMemoria(testando123).
true.

4 ?- palavra(testando123).
true.

5 ?- deletaPalavraNaMemoria(testando123).
true.

6 ?- palavra(testando123).
false.
```

Para que mesmo após o desligamento do programa, uma nova palavra continue registrada no banco, foi criado o predicado insereNovaPalavraNoDocumento(Palavra) , foi criado. Ele é responsável por escrever manualmente o fato com a nova palavra no próprio documento de palavras. Nota-se que para efeito imediato, é necessário também a inserção da palavra utilizando o predicado inserePalavraNaMemoria(Palavra) , que usa o assertz, para registrar tal fato.

4 Principais funcionalidades.

As principais funcionalidades disponíveis no programa são :

- *cifraCesar(Texto, Chave, TextoCifrado)* : Com este predicado conseguimos cifrar um Texto, passado pelo usuário, com uma devida Chave, também fornecida pelo mesmo. Em retorno, temos o TextoCifrado, que é a variável que possui o resultado desejado.

```
1 ?- cifraCesar('a ligeira raposa marrom saltou sobre o cachorro cansado' , 'c' , X).
X = "dcoljhludcudsrvdcpduurpcvdowrxcvreuhcrcfdfkruurcfdqvdgr" .
```

- *decifraCesar(Texto, Chave, TextoCifrado)* : Com este predicado conseguimos decifrar um TextoCifrado, passado pelo usuário, com uma devida Chave, também fornecida pelo mesmo. Em retorno, temos o Texto, que é a variável que possui o resultado desejado, ou seja, o texto original.

```
2 ?- decifraCesar(X , 'c' , 'dcoljhludcudsrvdcpduurpcvdowrxcvreuhcrcfdfkruurcfdqvdgr').
X = "a ligeira raposa marrom saltou sobre o cachorro cansado" .
```

- *cifraVigenere(Texto, Chave, TextoCifrado)* : Com este predicado conseguimos cifrar um Texto, passado pelo usuário, com uma devida Chave, também fornecida pelo mesmo. Em retorno, temos o TextoCifrado, que é a variável que possui o resultado desejado.

```
3 ?- cifraVigenere('flamengo campeao','lua',X).
X = "rGbyzosJaovnBzbA" .
```

- *decifraVigenere (Texto, Chave, TextoCifrado)* : Com este predicado conseguimos decifrar um TextoCifrado, passado pelo usuário, com uma devida Chave, também fornecida pelo mesmo. Em retorno, temos o Texto, que é a variável que possui o resultado desejado, ou seja, o texto original.

```
4 ?- decifraVigenere(X,'lua','rGbyzosJaovnBzbA').
X = "flamengo campeao" .
```

5 Dificuldades encontradas.

O maior problema que tive foi em relação a tempo para executar o trabalho. Logo ao tentar implementar os primeiros predicados percebi que precisava estudar mais a linguagem e fiquei uma semana só estudando. Além disso, contava com uma dupla para realizar as tarefas, só que por motivos pessoais, minha parceira teve que trancar a disciplina antes mesmo de começarmos a escrever qualquer código.

As primeiras dificuldades técnicas começaram logo depois de definir os predicados básicos como code e palavra. Dar o primeiro passo foi muito difícil pois o primeiro predicado mais complexo que tentei fazer foi o string2code, que é necessário em vários outros predicados. Com o uso de algumas funções já do próprio prolog como, string_chars e principalmente maplist, consegui concluir melhor essa tarefa.

A principal dificuldade que tive foi ao tentar implementar a quebra da cifra de César, tentei criar uma lista de caracteres mais usados na língua portuguesa em ordem e ir tentando decifrar com cada caractere e buscar a palavra na base de fatos, mas não consegui a tempo.

6 Execução do programa.

O trabalho pode ser executado a partir do seguinte comando :

$ swipl main.pl

Para o teste das principais funcionalidades, podemos utilizar os comandos :

$ cifraCesar('a ligeira raposa marrom saltou sobre o cachorro cansado' , 'c' , X).

$ decifraCesar(X , 'c' , 'dcoljhludcudsrvdcpduurpcvdowrxcvreuhcrcfdfkruurcfdqvdgr').

$ cifraVigenere('flamengo campeao' , 'lua' , X).

$ decifraVigenere(X, 'lua' , 'rGbyzosJaovnBzbA').

$ string2code('abcdef',X).

$ string2code(X,[1,2,3,4,5,6]).