

Relatório ep2 de MAC0425

Eduardo Dias Filho - NUSP: 7278100

&

Renato Scaroni - NUSP: 7289120

Novembro de 2013

Capítulo 1

Implementação

1.1 Geral

Neste relatório descreveremos detalhes de implementação do segundo exercício programa da disciplina MAC425, Inteligência artificial, ministrada no segundo semestre de 2013 para o curso de ciência da computação em conjunto com graduação e pós. Para esse programa foi utilizado SWI-PROLOG, como exigido no enunciado do exercício. Também seguindo orientações, a implementação foi dividida em 3 partes.

1.2 Parte 1

Esta parte consistia em extrair de uma frase o verbo, o núcleo do sujeito e o predicado e salvar isso em predicados lógicos. Para isso fizemos uso de uma gramática livre de contexto em prolog, que quebra uma frase em sujeito e predicado. Um sujeito, por sua vez, pode ser um artigo e um substantivo, ou um artigo e dois substantivos (respectivamente se falamos do fulano ou do professor fulano). Um predicado é um verbo seguido de um objeto, que por sua vez é um substantivo (pelomenos nos casos considerados). Os Símbolos iniciais da linguagem são S1p, S2 ou composta. S1p representa uma frase com a estrutura descrita acima com o verbo "da" e terminada com ponto final. S2 é uma frase na mesma estrutura de S1p, mas com o verbo "eh" e terminada em ponto final também. Composta é uma frase igual a S1p, porém sem o ponto final (definida pelo símbolo S1) seguida de uma frase S2. Esses símbolos iniciais representam uma sentença verificada pelo predicado `sentenca/3`

Para implementar os predicados que compõe a gramática descrita acima, foram implementadas os seguintes predicados auxiliares:

- **`ultimo_elem/2`** - Verifica se o primeiro parametro é o ultimo elemento da lista passada no segundo.
- **`ultimo_antes/3`** - Verifica se o primeiro parametro é o ultimo elemento da lista, passada no terceiro parametro, antes da ocorrência do elemento passado no segundo parametro.
- **`dias/2`** - Verifica se o dia passado no primeiro parametro é o n-esimo da semana, onde n é passado no segundo parametro.
- **`cp/2`** - Copia a lista no primeiro parametro para o segundo.
- **`eq/2`** - Verifica se os dois elementos passados por parametro são iguais
- **`corta_antes/3`** - Retorna no primeiro parametro tudo o que ocorre, em ordem, em uma lista, passada no segundo parâmetro, depois do elemento X, passado no terceiro.

Além desses predicados, foram usados 3 predicados dynamic, `eh/2`, `da/2` (guardam verbo, núcleo do sujeito e núcleo do predicado) e `genero/2`, que, quando o sujeito for um professor(a), guarda se é homem ou mulher. Para definir instâncias desse predicado utilizamos o predicado `assert/1`.

1.3 Parte 2

Essa parte consiste em ler a entrada do usuário através do predicado `monta_lista(X)` e transformar uma frase numa lista de palavras cujo último elemento é um ponto. Para isso utilizamos os seguintes predicados da linguagem: `read_line_to_codes/2`, `string_to_atom/2`, `downcase_atom/2` e `atomic_list_concat/2`. Além destes implementamos mais um predicado que utiliza o predicado `atom_concat/3`, para achar o fim da entrada e um predicado `final/1` que indica o fim da entrada.

1.4 Parte 3

Na ultima parte utilizamos as outras duas para realizar leitura da entrada e para acrescentar os predicados no caso de uma afirmação. Tivemos portanto que implementar o predicado `programa/1`, que executa o programa e definir a parte da busca e de formatação da saída. Para identificar uma busca usamos a gramática `query`, que, baseado no advérbio utilizado no início da frase (quem, quando e o que) decide que tipo de pergunta está sendo feita e qual predicado verificar ("da" ou "eh", definidos na parte 1). A estrutura da pergunta é quebrada de forma a identificar o verbo sobre o que está se perguntando (isto é se queremos saber quem dá qual matéria ou quando é uma matéria) através de ua gramática. Para gerar a saída de uma busca para usuário utilizamos o predicado `formata_saída/3` que se encarrega de montar a estrutura da frase de acordo com o resultado da busca ao banco de dados e depois utilizamos o predicado `write/1` para escrever na saída do prolog. Para decidir o genero de um professor na hora de montar a saída foi utilizado o predicado `genero/2` definido na parte 1.

Capítulo 2

Testes

2.1 Geral

Cada parte foi implementada e testada isoladamente, bem como cada predicado implementado. Utilizamos sempre entradas baseadas nas entradas dadas de exemplo no enunciado, ou seja, "professora renata da mac425.", no caso do ultimo predicado da parte 3, ou esta frase em forma de lista na parte 1. Para a montagem da lista utilizamos diversas frases, mesmo que suas estruturas não seguissem as das frases esperadas na entrada. Tudo funcionou muito bem. A única ressalva a ser feita é para o casoda busca, que exige que o enter seja pressionado antes da frase da pesquisa ser digitada, e que o programa só aceita afirmações com até dois dias para ocorrência de cada disciplina.