



Universidade do Minho
Licenciatura em Engenharia Informática

Inteligência Artificial

Relatório

Trabalho Prático 1ª Fase

André Gonçalves Pinto - A93173

Rui Pedro Alves - A93252

Tiago Pinheiro da Silva - A93285



Índice

1. Introdução	3
2. Base de conhecimento	3
3. Funcionalidades.....	4
3.1. Query 1	4
3.2. Query 2	4
3.3. Query 3	4
3.4. Query 4	5
3.5. Query 5	5
3.6. Query 6	5
3.7. Query 7	6
3.8. Query 8	6
3.9. Query 9	6
3.10. Query 10	7
4. Conclusão	7

1. Introdução

Este relatório foi realizado no âmbito da primeira fase do projeto da UC Inteligência Artificial, no qual nos é pedido para implementar um sistema de representação de conhecimento com capacidade para caracterizar um universo de discurso na área da logística de distribuição de encomendas.

Nesta primeira fase de entrega, recorrendo à linguagem de programação PROLOG, implementamos uma base de conhecimento e também algumas funcionalidades relativas ao problema apresentado.

2. Base de conhecimento

A coleção de predicados em um programa Prolog compõem coletivamente o que é conhecido como a base de conhecimento, assim sendo posteriormente à análise detalhada do problema proposto achamos necessários os seguintes factos e regras:

- **estafeta:** (Nome) -> {V,F}
Representamos o estafeta apenas pelo seu nome.
- **cliente:** (Nome | Morada) -> {V,F}
Representamos o cliente pelo seu nome e pela sua morada de residência.
- **encomenda:** (Objeto | Preço | Peso) -> {V,F}
Cada encomenda possui um objeto que é utilizado como identificador da encomenda, o seu preço e peso.
- **veículo:** (Nome | Nível de ecologia | Peso máximo | Velocidade média) -> {V,F}
Representamos o veículo pelo seu nome, Nível de ecologia, peso máximo que pode transportar e a sua velocidade média.
- **pedido:** (ID_pedido | Cliente | Distancia | Nome da encomenda | Hora | Dia | Mês | Ano | Horas para Entregar) -> {V,F}
Cada pedido é representado pelo id de pedido, cliente associado ao pedido, distância para entrega, encomenda associada, data em que foi realizado (hora, dia, mês e ano) e tempo de entrega em horas que o cliente escolhe para receber o seu pedido.
- **entrega:** (ID_Pedido | Estafeta | Cliente | Veículo | Encomenda | Peso | Hora | Dia | Mês | Ano | Rating) -> {V,F}

Uma entrega é referente a um pedido que foi devidamente entregue, esta é caracterizada por o id do respetivo pedido, nome do estafeta que a realizou, nome do cliente, veículo utilizado pelo estafeta, nome da encomenda, peso da encomenda, data em que foi entregue (hora, dia, mês e ano) e o respetivo rating dado pelo cliente.

- **mes:** (Nome, Número) -> {V,F}
Acrescentamos também um parâmetro mês com o nome do mês e o seu respetivo número, para facilitar a implementação de algumas funcionalidades.

3. Funcionalidades

3.1. Query 1

Q1: “Identificar o estafeta que utilizou mais vezes um meio de transporte mais ecológico”

Para realizar esta query foi necessário saber o nível de ecologia de um determinado estafeta. Dessa forma atribuímos pontos ao estafeta por usar determinado veículo, 3 pontos por utilizar bicicleta, 2 para moto e 1 para carro, sendo o seu nível de ecologia calculado através da soma de todos pontos que obteve a dividir pelo número de entregas realizado. Assim, o estafeta com maior pontuação é o mais ecológico.

Tendo a pontuação de todos os estafetas calculadas criamos um predicado recursivo usando um acumulador do estafeta mais ecológico. Desta maneira a cada iteração do predicado temos como argumento inicial o estafeta mais ecológico e o resto da lista de estafetas, sendo que no final deste predicado descobrimos o estafeta mais ecológico.

3.2. Query 2

Q2: “identificar que estafetas entregaram determinada(s) encomenda(s) a um determinado cliente”

Primeiramente utilizamos um predicado auxiliar “qualEstafeta” que dado um Cliente e uma Encomenda, faz pattern matching nas entregas e retira o respetivo Estafeta associado, em seguida no predicado “query2” é devolvida a lista de estafetas (sem ocorrências repetidas pois utilizamos “setof” para criar a lista) associada à encomenda e cliente escolhido.

3.3. Query 3

Q3: “identificar os clientes servidos por um determinado estafeta”

O predicado auxiliar “qualCliente” dado um estafeta faz pattern matching nas entregas e devolve o cliente associado.

No predicado “query3” dado um determinado estafeta e recorrendo ao predicado auxiliar “qualCliente” é devolvida a lista de clientes, sem ocorrências repetidas (utilizando “setof”), aos quais o estafeta realizou entregas.

3.4. Query 4

Q4: “calcular o valor faturado pela Green Distribution num determinado dia

No predicado “faturacaoAux” dado uma data (dia, mês e ano) esta faz pattern matching da mesma data nas entregas e retirando a respetiva encomenda, em seguida faz novamente pattern matching da encomenda e retira o respetivo preço.

O predicado “query4” recebendo uma data cria uma lista, recorrendo ao predicado auxiliar “faturacaoAux”, com o preço de todas as encomendas entregues desse dia e em seguida soma todos os elementos da lista devolvendo o valor faturado nesse dia.

3.5. Query 5

Q5: “identificar quais as zonas (e.g., rua ou freguesia) com maior volume de entregas por parte da Green Distribution”

Primeiramente guardamos numa lista, ruas associadas a entregas. Nessa lista o número de ocorrências de uma determinada rua é o número de vezes que foi feita uma entrega a essa rua.

Para comparar qual a rua teve mais entregas chamamos um predicado que devolve o número de ocorrências de um determinado elemento na lista na lista.

Após chamar o predicado recursivamente obtemos a rua com maior volume de entregas.

3.6. Query 6

Q6: “calcular a classificação média de satisfação de cliente para um determinado estafeta”

O predicado auxiliar “ratingEstafeta” faz pattern matching do estafeta inserido nas entregas e devolve o rating deste.

O predicado “query6” recebendo um estafeta cria uma lista, com o auxílio do predicado “ratingEstafeta”, com todos os ratings associados a esse estafeta e devolve a média desta lista.

Para esta query foram criados também os predicados recursivos para obter a soma dos elementos de uma lista e obter o tamanho de uma lista.

3.7. Query 7

Q7: “identificar o número total de entregas pelos diferentes meios de transporte, num determinado intervalo de tempo”

Primeiramente utilizamos o predicado mais simples que calcula para um dado veículo se as suas entregas foram feitas, dentro do período de tempo pré-estabelecido.

Usando o predicado findall, no predicado “dataEntregaVeiculo”, obtemos para um dado veículo as suas entregas totais.

Chamando agora o predicado findall, no predicado “allEntregas”, obtemos numa lista o número de entregas totais para cada tipo de veículo. Somando essa lista obtemos o número total de entregas para um determinado período de tempo.

3.8. Query 8

Q8: “identificar o número total de entregas pelos estafetas, num determinado intervalo de tempo”

Criamos o predicado mais simples que calcula, para um dado estafeta, se as suas entregas foram feitas dentro do período de tempo pré-estabelecido.

De seguida, usando o predicado findall no predicado “dataEntregaEstafeta”, obtemos todas as entregas de um determinado estafeta.

Por fim, chamando o findall no predicado “query8aux”, conseguimos obter uma lista com o número de entregas de cada estafeta, sendo a soma desta lista o número total de entregas pelos estafetas.

3.9. Query 9

Q9: “calcular o número de encomendas entregues e não entregues pela Green Distribution, num determinado período de tempo”

Verificamos se um pedido está dentro do período de tempo com recurso ao predicado “dentroTempo”, sendo que depois verificamos se esse pedido foi entregue, com recurso ao predicado “foiEntregue”.

Sabendo que o pedido foi entregue este é adicionado a uma lista, sendo os pedidos que não verificam esta condição adicionados a uma outra lista.

A soma dos tamanhos de ambas as listas retorna o número de encomendas entregues e não entregues pela Green Distribution.

3.10. Query 10

Q10: “calcular o peso total transportado por estafeta num determinado dia”

O predicado “pesoEstafeta” dado um determinado estafeta e uma data (dia, mês e ano) faz pattern matching destes dados nas entregas retirando a respetiva encomenda e posteriormente pattern matching do nome da encomenda nas encomendas retirando o respetivo peso.

No predicado “query10” dado o dia e o estafeta, é criada a lista com o peso de todas as encomendas realizadas nesse dia por esse mesmo estafeta, no final é devolvida a soma de todos os elementos dessa lista, obtendo assim o peso total transportado nesse dia.

4. Conclusão

Nesta primeira fase do trabalho definimos uma base de conhecimento com fatos e regras essenciais à implementação das funcionalidades pretendidas, podendo esta, em fases futuras, ser melhorada, indo ao encontro de eventuais novas funcionalidades e objetivos.

Em suma, pensamos ter atingido todos os objetivos propostos nesta fase inicial, sendo o trabalho desenvolvido até ao momento fundamental para o desenvolvimento futuro do projeto.