

1. Base de conhecimento

O sistema de representação de conhecimento e raciocínio desenvolvido caracteriza a logística de distribuição de encomendas. O conhecimento é dado pelo seguinte modo:

- cliente: #IdCliente, Morada -> {V,F}
 - estafeta: #IdEstafeta, Nome-> {V,F}
 - encomenda: #IdEncomenda, Peso, Volume -> {V,F}
 - entrega: Data, #IdEntrega, #IdCliente, #IdEstafeta, , MeioTransporte, Custo -> {V,F}
-

Base de conhecimento inicial:

Para a realização do trabalho, é necessário que o programa já possua uma base de conhecimento inicial, de maneira a que possamos testar a nossa informação, sem termos de inserir conhecimento.

Criação de clientes/estafetas/encomendas/entregas:

1.1. cliente

Caracterizado por um identificador e uma morada. Os seguintes predicados representam clientes:

cliente(0, 'Rua do Meio').
cliente(1, 'Rua do Verde').
cliente(2, 'Rua do Amarelo').

1.2. estafeta

Caracterizado por um identificador e um nome. Os seguintes predicados representam estafetas:

estafeta(0, 'Daniel').
estafeta(1, 'Nuno').
estafeta(2, 'Guilherme').
estafeta(3, 'Rodrigo').

1.3. encomenda

Caracterizada por identificador de encomenda, peso e volume. Os seguintes predicados representam encomendas:

encomenda(0, 2.5, 10).
encomenda(1, 5.2, 15).
encomenda(2, 19.1, 30).

1.4. entrega

Caracterizada por Data, identificador de entrega, identificador de cliente, identificador de estafeta, meio de transporte e custo. Os seguintes predicados representam entregas:.

entrega(data(13, 12, 2021), 1, 2, 4, 'Bicicleta', 19.99).
entrega(data(16, 11, 2021), 2, 9, 3, 'Mota', 29.99).
entrega(data(18, 09, 2021), 3, 5, 2, 'Carro', 39.99).
entrega(data(19, 05, 2021), 4, 8, 4, 'Bicicleta', 19.99).

O predicado **solucoes**

solucoes(F,Q,S) :- findall(F, Q, S).

Evolução e involução

evolução: adicionar conhecimento à Base de Conhecimento

involução: remover conhecimento da Base

Invariantes

Para evitar problemas com a Base de Conhecimento e para garantir que, no momento de inserção/remoção de conhecimento, haja controlo dessa informação, usamos invariantes.

Resumindo:

Caso um invariante não se verifique, os predicados **evolução** e **involução** impedem qualquer ação. Assim, garantimos a consistência da nossa **Base de Conhecimento**.

Invariantes associados à inserção(prefixo "+"):

- impedir repetição de conhecimento (verificar se ID do predicado é único, caso contrário remove da Base)
- assegurar que não é possível inserir entrega cujos id de cliente/estafeta não existam na base de conhecimento

Invariantes associados à remoção(prefixo "-")

