TP6 - Machine de Turing

Paul Chaignon - Clément Gautrais

November 27, 2013

1 Bases de données déductives

1.1 Questions

Listing 1: baseauto.pro

```
1  /**
2  TP 7 Base de DonnÃ©es DÃ©ductives (BDD) - Prolog
3
4  @author Paul Chaignon
5  @author ClÃ©ment ClÃ©ment
6  @version Annee scolaire 2013/2014
7  */
8
9
10 /*
11 ================================================================================
12 ================================================================================
13  DÃ©finition des prÃ©dicats
14 ================================================================================
15 */
16 %
     ================================================================================
17 % SECTION 1 : Base de donnÃ©es
18 %
     ================================================================================
19
20 assemblage(voiture, porte, 4).
21 assemblage(voiture, roue, 4).
22 assemblage(voiture, moteur, 1).
23 assemblage(roue, jante, 1).
24 assemblage(porte, tole, 1).
25 assemblage(porte, vitre, 1).
26 assemblage(roue, pneu, 1).
27 assemblage(moteur, piston, 4).
28 assemblage(moteur, soupape, 16).
29
30
31 piece(p1, tole, lyon).
32 piece(p2, jante, lyon).
33 piece(p3, jante, marseille).
34 piece(p4, pneu, clermontFerrand).
35 piece(p5, piston, toulouse).
36 piece(p6, soupape, lille).
```

```
37 piece(p7, vitre, nancy).
38 piece(p8, tole, marseille).
39 piece(p9, vitre, marseille).
40
41
42 demandeFournisseur(dupont, lyon).
43 demandeFournisseur(michel, clermontFerrand).
44 demandeFournisseur(durand, lille).
45 demandeFournisseur(dupond, lille).
46 demandeFournisseur(martin, rennes).
47 demandeFournisseur(smith, paris).
48 demandeFournisseur(brown, marseille).
49
50
51 fournisseurReference(f1, dupont, lyon).
52 fournisseurReference(f2, durand, lille).
53 fournisseurReference(f3, martin, rennes).
54 fournisseurReference(f4, michel, clermontFerrand).
55 fournisseurReference(f5, smith, paris).
56 fournisseurReference(f6, brown, marseille).
57
58
59 livraison(f1, p1, 300).
60 livraison(f2, p2, 200).
61 livraison(f3, p3, 200).
62 livraison(f4, p4, 400).
63 livraison(f6, p5, 500).
64 livraison(f6, p6, 1000).
65 livraison(f6, p7, 300).
66 livraison(f1, p2, 300).
67 livraison(f4, p2, 300).
68 livraison(f4, p1, 300).
69
70
71 %
     ================================================================================
72 % SECTION 2 : Operation relationnelles
73 %
     ================================================================================
74
75 %fromcity(+City, -piece(NumPiece, Nom, City))
76 fromcity(City, piece(NumPiece, Nom, City)):-
77   piece(NumPiece, Nom, City).
78
79 %infospieces(?Nom, ?Lieu)
80 infospieces(Nom, Lieu):-
81   piece(_, Nom, Lieu).
82
83 %inter(?Nom, ?Lieu)
84 inter(Nom, Lieu):-
85   demandeFournisseur(Nom, Lieu),
86   fournisseurReference(_, Nom, Lieu).
87
88 %union(?Nom, ?Lieu)
89 union(Nom, Lieu):-
90   demandeFournisseur(Nom, Lieu).
91 union(Nom, Lieu):-
92   fournisseurReference(_, Nom, Lieu),
```

```
93    not(demandeFournisseur(Nom, Lieu)).
94
95  %diff(?Nom, ?Lieu)
96  diff(Nom, Lieu):-
97    demandeFournisseur(Nom, Lieu),
98    not(fournisseurReference(_, Nom, Lieu)).
99
100 %prodcart(-uplet(NumFourn1, Nom, Ville, NumFourn2, Piece, Quantite))
101 prodcart(uplet(NumFourn1, Nom, Ville, NumFourn2, Piece, Quantite)):-
102   fournisseurReference(NumFourn1, Nom, Ville),
103   livraison(NumFourn2, Piece, Quantite).
104
105 %jointure(-uplet(NumFourn, Nom, Ville, Piece, Quantite))
106 jointure(uplet(NumFourn, Nom, Ville, Piece, Quantite)):-
107   fournisseurReference(NumFourn, Nom, Ville),
108   livraison(NumFourn, Piece, Quantite).
109
110 %jointureSelect(-uplet(NumFourn, Nom, Ville, Piece, Quantite))
111 jointureSelect(uplet(NumFourn, Nom, Ville, Piece, Quantite)):-
112   jointure(uplet(NumFourn, Nom, Ville, Piece, Quantite)),
113   Quantite > 350.
114
115 jointureSelect2(uplet(NumFourn, Nom, Ville, Piece, Quantite)):-
116   livraison(NumFourn, Piece, Quantite),
117   Quantite > 350,
118   fournisseurReference(NumFourn, Nom, Ville).
119
120 pasOk(Fourn1):-
121   piece(PieceLyon, _, lyon),
122   not(livraison(Fourn1, PieceLyon, _)).
123 %div(-Fourn)
124 div(Fourn):-
125   fournisseurReference(Fourn, _, _),
126   not(pasOk(Fourn)).
127
128 sumQuantites([], 0).
129 sumQuantites([Prem|List], Total):-
130   sumQuantites(List, Total2),
131   Total is Total2 + Prem.
132
133 totalPieces(Fourn, NB):-
134   findall(Quantite, livraison(Fourn, _, Quantite), Quantites),
135   sumQuantites(Quantites, NB).
136
137 %totalPieces(-uplet(Fourn, NB))
138 totalPieces(uplet(Fourn, NB)):-
139   fournisseurReference(Fourn, _, _),
140   totalPieces(Fourn, NB).
141
142
143
144 %
      ================================================================================
145 % SECTION 3 : Au dela de l algebre relationnelle
146 %
      ================================================================================
147
148 %Q1
```

```
149 %composant(+Composant, -ComposeDe)
150 composant(Composant, ComposeDe):-
151   assemblage(Composant, ComposeDe, _).
152
153 composant(Composant, ComposeDe):-
154   assemblage(Composant, SousComposant, _),
155   composant(SousComposant, ComposeDe).
156
157
158 % Q2
159
160 nbExemplaires(Composant, Piece, N):-
161   assemblage(Composant, Piece, N).
162
163 nbExemplaires(Composant, Piece, N):-
164   assemblage(Composant, ComposantInter, M),
165   nbExemplaires(ComposantInter, Piece, P),
166   N is M*P.
167
168 estPieceCut(P):-
169   piece(_, P, _),
170   !.
171
172 %piecesTotal(+Composant, -composant(R,N))
173 piecesTotal(Composant, composant(R,N)):-
174   composant(Composant, R),
175   estPieceCut(R),
176   nbExemplaires(Composant, R, N).
177
178
179 %Q3
180 sumQuantProd([], 0).
181
182 sumQuantProd([Prod|R], NB):-
183   nbComposantFourni(Prod, Nombre),
184   sumQuantProd(R, NbR),
185   NB is NbR+Nombre.
186
187 nbComposantFourniNom(Nom, NB):-
188   findall(CodeComp, piece(CodeComp, Nom, _), Prods),
189   sumQuantProd(Prods, NB).
190
191 nbComposantFourni(CodeComp, NB):-
192   findall(Quantite, livraison(_, CodeComp, Quantite), Quantites),
193   sumQuantites(Quantites, NB).
194
195 ratioComposant([], []).
196
197 ratioComposant([composant(NomComp,N)|Compo], [Rat1|Rest]):-
198   nbComposantFourniNom(NomComp, T),
199   Rat1 is T/N,
200   ratioComposant(Compo, Rest).
201
202 minBis([], Min, Min).
203
204 minBis([P|R], Min, Res):-
205   P < Min,
206   !,
```

```
207   minBis(R, P, Res).
208
209 minBis([P|R], Min, Res):-
210   minBis(R, Min, Res).
211
212 minL([P|R], N):-
213   minBis(R, P, N).
214
215 %nbVoit(-NB)
216 nbVoit(NB):-
217   findall(composant(Comp,N), piecesTotal(voiture, composant(Comp, N)),
          Composants),
218   ratioComposant(Composants, Ratios),
219   minL(Ratios, NBfloat),
220   NB is truncate(NBfloat).
```

1.2 Tests

Listing 2: baseauto_tests.pro

```
1  /*
2  ================================================================================
3  ================================================================================
4   Tests
5  ================================================================================
6  */
7
8  /*
9  Q2.1
10
11 fromcity(lyon, R).
12
13 R = piece(p1, tole, lyon)
14 Yes (0.00s cpu, solution 1, maybe more) ? ;
15
16 R = piece(p2, jante, lyon)
17 Yes (0.00s cpu, solution 2)
18 */
19
20 /*
21 Q2.2
22 infospieces(N, L).
23
24 N = tole
25 L = lyon
26 Yes (0.00s cpu, solution 1, maybe more) ? ;
27
28 N = jante
29 L = lyon
30 Yes (0.00s cpu, solution 2, maybe more) ? ;
31
32 N = jante
33 L = marseille
34 Yes (0.00s cpu, solution 3, maybe more) ? ;
```

```
35
36 N = pneu
37 L = clermontFerrand
38 Yes (0.00s cpu, solution 4, maybe more) ?
39 ...
40 */
41
42 /*
43 Q2.3
44 inter(N, L).
45
46 N = dupont
47 L = lyon
48 Yes (0.00s cpu, solution 1, maybe more) ? ;
49
50 N = michel
51 L = clermontFerrand
52 Yes (0.00s cpu, solution 2, maybe more) ? ;
53
54 N = durand
55 L = lille
56 Yes (0.00s cpu, solution 3, maybe more) ? ;
57
58 N = martin
59 L = rennes
60 Yes (0.00s cpu, solution 4, maybe more) ? ;
61
62 N = smith
63 L = paris
64 Yes (0.00s cpu, solution 5, maybe more) ? ;
65
66 N = brown
67 L = marseille
68 Yes (0.00s cpu, solution 6)
69
70
71
72 union(N, L).
73
74 N = dupont
75 L = lyon
76 Yes (0.00s cpu, solution 1, maybe more) ? ;
77
78 N = michel
79 L = clermontFerrand
80 Yes (0.00s cpu, solution 2, maybe more) ? ;
81
82 N = durand
83 L = lille
84 Yes (0.00s cpu, solution 3, maybe more) ? ;
85
86 N = dupond
87 L = lille
88 Yes (0.00s cpu, solution 4, maybe more) ? ;
89
90 N = martin
91 L = rennes
92 Yes (0.00s cpu, solution 5, maybe more) ? ;
```

```
93
94 N = smith
95 L = paris
96 Yes (0.00s cpu, solution 6, maybe more) ? ;
97
98 N = brown
99 L = marseille
100 Yes (0.00s cpu, solution 7, maybe more) ? ;
101
102 No (0.00s cpu)
103
104
105 diff(N, L).
106
107 N = dupond
108 L = lille
109 Yes (0.00s cpu, solution 1, maybe more) ? ;
110
111 No (0.00s cpu)
112 */
113
114 /*
115 Q2.4
116
117 prodcart(R).
118
119 R = uplet(f1, dupont, lyon, f1, p1, 300)
120 Yes (0.00s cpu, solution 1, maybe more) ? ;
121
122 R = uplet(f1, dupont, lyon, f2, p2, 200)
123 Yes (0.01s cpu, solution 2, maybe more) ? ;
124
125 R = uplet(f1, dupont, lyon, f3, p3, 200)
126 Yes (0.01s cpu, solution 3, maybe more) ? ;
127
128 R = uplet(f1, dupont, lyon, f4, p4, 400)
129 Yes (0.01s cpu, solution 4, maybe more) ? ;
130
131 R = uplet(f1, dupont, lyon, f6, p5, 500)
132 Yes (0.01s cpu, solution 5, maybe more) ? ;
133
134 R = uplet(f1, dupont, lyon, f6, p6, 1000)
135 Yes (0.01s cpu, solution 6, maybe more) ? ;
136
137 R = uplet(f1, dupont, lyon, f6, p7, 300)
138 Yes (0.01s cpu, solution 7, maybe more) ? ;
139 ... (60 solutions)
140
141 */
142
143 /*
144 Q2.5
145 jointure(R).
146
147 R = uplet(f1, dupont, lyon, p1, 300)
148 Yes (0.00s cpu, solution 1, maybe more) ? ;
149
150 R = uplet(f1, dupont, lyon, p2, 300)
```

```
Yes (0.00s cpu, solution 2, maybe more) ? ;

R = uplet(f2, durand, lille, p2, 200)
Yes (0.00s cpu, solution 3, maybe more) ? ;

R = uplet(f3, martin, rennes, p3, 200)
Yes (0.01s cpu, solution 4, maybe more) ? ;

R = uplet(f4, michel, clermontFerrand, p4, 400)
Yes (0.01s cpu, solution 5, maybe more) ? ;

R = uplet(f4, michel, clermontFerrand, p2, 300)
Yes (0.01s cpu, solution 6, maybe more) ? ;

R = uplet(f4, michel, clermontFerrand, p1, 300)
Yes (0.01s cpu, solution 7, maybe more) ? ;

R = uplet(f6, brown, marseille, p5, 500)
Yes (0.01s cpu, solution 8, maybe more) ? ;

R = uplet(f6, brown, marseille, p6, 1000)
Yes (0.01s cpu, solution 9, maybe more) ? ;

R = uplet(f6, brown, marseille, p7, 300)
Yes (0.01s cpu, solution 10)



jointureSelect(R).

R = uplet(f4, michel, clermontFerrand, p4, 400)
Yes (0.00s cpu, solution 1, maybe more) ? ;

R = uplet(f6, brown, marseille, p5, 500)
Yes (0.00s cpu, solution 2, maybe more) ? ;

R = uplet(f6, brown, marseille, p6, 1000)
Yes (0.00s cpu, solution 3, maybe more) ? ;

No (0.00s cpu)
*/

/*
Q2.6
div(F).

F = f1
Yes (0.00s cpu, solution 1, maybe more) ? ;

F = f4
Yes (0.00s cpu, solution 2, maybe more) ? ;

No (0.00s cpu)
*/

/*
Q2.7
totalPieces(R).
```

```
209
210 R = uplet(f1, 600)
211 Yes (0.00s cpu, solution 1, maybe more) ? ;
212
213 R = uplet(f2, 200)
214 Yes (0.00s cpu, solution 2, maybe more) ? ;
215
216 R = uplet(f3, 200)
217 Yes (0.00s cpu, solution 3, maybe more) ? ;
218
219 R = uplet(f4, 1000)
220 Yes (0.00s cpu, solution 4, maybe more) ? ;
221
222 R = uplet(f5, 0)
223 Yes (0.00s cpu, solution 5, maybe more) ? ;
224
225 R = uplet(f6, 1800)
226 Yes (0.00s cpu, solution 6)
227 */
228
229 /*
230 Q3.1
231 composant(voiture, R).
232
233 R = porte
234 Yes (0.00s cpu, solution 1, maybe more) ? ;
235
236 R = roue
237 Yes (0.00s cpu, solution 2, maybe more) ? ;
238
239 R = moteur
240 Yes (0.00s cpu, solution 3, maybe more) ? ;
241
242 R = tole
243 Yes (0.00s cpu, solution 4, maybe more) ? ;
244
245 R = vitre
246 Yes (0.00s cpu, solution 5, maybe more) ? ;
247
248 R = jante
249 Yes (0.00s cpu, solution 6, maybe more) ? ;
250
251 R = pneu
252 Yes (0.00s cpu, solution 7, maybe more) ? ;
253
254 R = piston
255 Yes (0.00s cpu, solution 8, maybe more) ? ;
256
257 R = soupape
258 Yes (0.00s cpu, solution 9, maybe more) ? ;
259
260 No (0.00s cpu)
261 */
262
263 /*
264 Q3.2
265 piecesTotal(voiture, R).
266
```

```
R = composant(tole, 4)
Yes (0.00s cpu, solution 1, maybe more) ? ;

R = composant(vitre, 4)
Yes (0.00s cpu, solution 2, maybe more) ? ;

R = composant(jante, 4)
Yes (0.00s cpu, solution 3, maybe more) ? ;

R = composant(pneu, 4)
Yes (0.00s cpu, solution 4, maybe more) ? ;

R = composant(piston, 4)
Yes (0.00s cpu, solution 5, maybe more) ? ;

R = composant(soupape, 16)
Yes (0.00s cpu, solution 6, maybe more) ? ;

No (0.00s cpu)
*/

/*
Q3.3
nbVoit(NB).

NB = 62.0
Yes (0.00s cpu)
*/
```