

TP3 - Liste

PAUL CHAIGNON - CLÉMENT GAUTRAIS

October 3, 2013

1 Questions

Listing 1: tp_listes.pro

```
1 /**
2  * membre(?A, +X)
3  */
4 membre(A, [A|R]).
5 membre(A, [X|R]):-
6     membre(A, R).
7
8 /**
9  * compte(+A, +X, ?N)
10 */
11 compte(A, [], 0).
12 compte(A, [A|R], N):-
13     compte(A, R, M),
14     N is M + 1.
15 compte(A, [X|R], N):-
16     A \== X,
17     compte(A, R, N).
18
19 /**
20 * renverser(+X, ?Y)
21 */
22 renverser(X, Y):-
23     renv(X, [], Y).
24 renv([X|R], A, Y):-
25     renv(R, [X|A], Y).
26 renv([], Y, Y).
27
28 /**
29 * palind(+X)
30 */
31 palind(X):-
32     renverser(X, X).
33
34 /**
35 * nieme1(+N, +X, -A)
36 */
37 nieme1(0, [X|R], X).
38 nieme1(N, [Y|R], X):-
39     N \== 0,
40     M is N - 1,
```

```

41     nieme1(M, R, X).
42
43 /**
44  * nieme2(-N, +X, +A)
45  */
46 nieme2(0, [X|R], X).
47 nieme2(N, [Y|R], X):-
48     X \== Y,
49     nieme2(M, R, X),
50     N is M + 1.
51 /**
52  * Pas possible d'avoir un algo commun car ils ne font pas la m me
53   chose.
54  * De plus, il y aurait des probl mes si une liste contient 2 valeurs
55   identique.
56  */
57 /**
58  * hors_de(+A, +X)
59  */
60 hors_de(A, X):-
61     compte(A, X, 0).
62 /**
63  * tous_diff(+X)
64  */
65 tous_diff([]).
66 tous_diff([X|R]):-
67     hors_de(X, R),
68     tous_diff(R).
69 /**
70  * conc3(+X, +Y, +Z, ?T)
71  */
72 conc3([], [], Z, Z).
73 conc3([], [P|R], Z, [P|T]):-
74     conc3([], R, Z, T).
75 conc3([P|R], Y, Z, [P|T]):-
76     conc3(R, Y, Z, T).
77 /**
78  * Oui c'est possible et c'est le cas de notre algorithme.
79  */
80 /**
81  * debute_par(+X, ?Y)
82  */
83 debute_par(X, []).
84 debute_par([X|R], [X|Q]):-
85     debute_par(R, Q).
86 /**
87  * sous_liste(+X, ?Y)
88  */
89 sous_liste(X, Y):-
90     debute_par(X, Y).
91 sous_liste([X|R], Y):-
92     sous_liste(R, Y).
93
94
95
96

```

```

97 /**
98  * elim(+X, -Y)
99  */
100 elim(X, Y):-
101     elimin(X, Y, []).
102 elimin([], Z, Z).
103 elimin([X|R], Y, Z):-
104     compte(X, Z, 1),
105     elimin(R, Y, Z).
106 elimin([X|R], Y, Z):-
107     compte(X, Z, 0),
108     elimin(R, Y, [X|Z]).
109
110 /**
111  * inserer(+E, +L1, -L2)
112  */
113 inserer(E, [], [E]).
114 inserer(E, [P|R], [E,P|R]):-
115     E =< P.
116 inserer(E, [P|R], [P|Z]):-
117     E > P,
118     inserer(E, R, Z).
119
120 /**
121  * tri(+X, -Y)
122  */
123 tri(X, Y):-
124     trier(X, Y, []).
125 trier([], Acc, Acc).
126 trier([X|R], Y, Acc):-
127     inserer(X, Acc, NewAcc),
128     trier(R, Y, NewAcc).
129
130 /**
131  * inclus(+X, +Y)
132  */
133 inclus([], Y).
134 inclus([X|R], Y):-
135     membre(X, Y),
136     inclus(R, Y).
137
138 /**
139  * non_inclus(+X, +Y)
140  */
141 non_inclus([X|R], Y):-
142     hors_de(X, Y).
143 non_inclus([X|R], Y):-
144     membre(X, Y),
145     non_inclus(R, Y).
146
147 /**
148  * Le cut dans les 3 fonctions suivantes servent à g nerer des
149     ensembles sans doublons.
150  */
151 /**
152  * union_ens(+X, +Y, ?Z)
153  */
154 union_ens(X, Y, Z):-

```

```

154 union_ensem(X, Y, Z, []).
155 union_ensem([], [], Z, Acc):-
156     inclus(Acc, Z),
157     inclus(Z, Acc),
158     !.
159 union_ensem([X|R], Y, Z, Acc):-
160     hors_de(X, Acc),
161     union_ensem(R, Y, Z, [X|Acc]).
162 union_ensem([X|R], Y, Z, Acc):-
163     membre(X, Acc),
164     union_ensem(R, Y, Z, Acc).
165 union_ensem([], [Y|R], Z, Acc):-
166     hors_de(Y, Acc),
167     union_ensem([], R, Z, [Y|Acc]).
168 union_ensem([], [Y|R], Z, Acc):-
169     membre(Y, Acc),
170     union_ensem([], R, Z, Acc).
171
172 /**
173  * inter_ens(+X, +Y, ?Z)
174  */
175 inter_ens(X, Y, Z):-
176     inter_ensem(X, Y, Z, []).
177 inter_ensem([], Y, Z, Acc):-
178     inclus(Acc, Z),
179     inclus(Z, Acc),
180     !.
181 inter_ensem([X|R], Y, Z, Acc):-
182     membre(X, Y),
183     inter_ensem(R, Y, Z, [X|Acc]).
184 inter_ensem([X|R], Y, Z, Acc):-
185     hors_de(X, Y),
186     inter_ensem(R, Y, Z, Acc).
187
188 /**
189  * diff_ens(+X, +Y, ?Z)
190  */
191 diff_ens(X, Y, Z):-
192     diff_ensem(X, Y, Z, []).
193 diff_ensem([], Y, Z, Acc):-
194     inclus(Acc, Z),
195     inclus(Z, Acc),
196     !.
197 diff_ensem([X|R], Y, Z, Acc):-
198     hors_de(X, Y),
199     diff_ensem(R, Y, Z, [X|Acc]).
200 diff_ensem([X|R], Y, Z, Acc):-
201     membre(X, Y),
202     diff_ensem(R, Y, Z, Acc).

```

2 Tests

Listing 2: tp_listes_tests.pro

```

1 membre(2, [1,2,3,4]). % Yes
2 membre(5, [1,2,3,4]). % No

```

```

3
4 compte(1, [1,2,3,4], X). % X = 1
5 compte(5, [1,2,3,4], X). % X = 0
6 compte(1, [1,2,1,4], X). % X = 2
7 compte(1, [1,2,1,4], 2). % Yes
8
9 renverser([1,2,3,4], Y). % Y = [4,3,2,1]
10 renverser([1,2,3,4], [4,3,2,1]). % Yes
11
12 palind([1,2,3,4]). % No
13 palind([4,3,3,4]). % Yes
14 palind([4]). % Yes
15
16 nieme(0, [1,2,3,4], A). % A = -1
17 nieme(3, [1,2,3,4], A). % A = 2
18
19 hors_de(0, [1,2,3,4]). % Yes
20 hors_de(3, [1,2,3,4]). % No
21
22 tous_diff([1,2,3,4]). % Yes
23 tous_diff([1,2,4,4]). % No
24 tous_diff([]). % Yes
25
26 conc3([1,2], [3,4], [5,6], T). % T = [1,2,3,4,5,6]
27 conc3([1,2], [3,4], [5,6], [1,2,3,4,5,6]). % Yes
28 conc3([1,2], [3,4], [5,6], [1,2,3,4,5]). % No
29 conc3([], [], [], []). % Yes
30
31 debute_par([1,2,3], []). % Yes
32 debute_par([1,2,3], [1]). % Yes
33 debute_par([1,2,3], [1,2,3]). % Yes
34 debute_par([1,2,3], [2,3]). % No
35
36 sous_liste([1,2,3], []). % Yes
37 sous_liste([1,2,3], [1]). % Yes
38 sous_liste([1,2,3], [2,3]). % Yes
39 sous_liste([1,2,3], [1,3]). % No
40
41 elim([1,2,3], Y). % Y = [1,2,3]
42 elim([3,1,2,3,3], Y). % Y = [1,2,3]
43 elim([], Y). % Y = []
44
45 tri([1,2,3,4], Y). % Y = [1,2,3,4]
46 tri([4,3,2,1], Y). % Y = [1,2,3,4]
47 tri([3,1,2,4], Y). % Y = [1,2,3,4]
48
49 inclus([], [1,2,3,4]). % Yes
50 inclus([1,2,3,4], [1,2,3,4]). % Yes
51 inclus([1,2,4], [1,2,3,4]). % Yes
52 inclus([1,2,5], [1,2,3,4]). % No
53
54 non_inclus([1,2,3,4], []). % Yes
55 non_inclus([5], [1,2,3,4]). % Yes
56 non_inclus([], [1,2,3,4]). % No
57 non_inclus([2,3], [1,2,3,4]). % No
58
59 union_ens([1,2,3], [2,3,4], [1,2,3,4]). % Yes
60 union_ens([2,3,1], [4,2,3], [3,2,1,4]). % Yes

```

```

61 union_ens([2,3,1], [4,2,3], [3,2,1]). % No
62 union_ens([1,2,3], [2,3,4], Z). % Z = [1,2,3,4]
63 union_ens([2,3,1], [4,2,3], Z). % Z = [1,2,3,4]
64
65 inter_ens([1,2,3], [2,3,4], [2,3]). % Yes
66 inter_ens([2,3,1], [4,2,3], [3,2]). % Yes
67 inter_ens([2,3,1], [4,2,3], [3,2,1]). % No
68 inter_ens([1,2,3], [2,3,4], Z). % Z = [2,3]
69 inter_ens([2,3,1], [4,2,3], Z). % Z = [2,3]
70
71 diff_ens([1,5,3], [5,4,7], Z). % Z = [1,3]
72 diff_ens([1,5,3], [], Z). % Z = [5,1,3]
73 diff_ens([1,5,3], [5,4,7], [3,1]). % Yes
74 diff_ens([1,5,3], [5,4,7], [1,3]). % Yes
75 diff_ens([1,5,3], [5,4,7], [1,3,5]). % No
76 diff_ens([1,5,3], [5,4,7], [1]). % No

```