

TP4 - Arbres binaires

PAUL CHAIGNON - CLÉMENT GAUTRAIS

October 5, 2013

1 Questions

Listing 1: tp_arbres.pro

```
1 /**
2  * Question 1
3  * arbre_binaire(+B)
4  */
5 arbre_binaire(arb_bin(R, G, D)):-
6     integer(R),
7     arbre_binaire(G),
8     arbre_binaire(D).
9 arbre_binaire(arb_bin(R, vide, D)):-
10     integer(R),
11     arbre_binaire(D).
12 arbre_binaire(arb_bin(R, G, vide)):-
13     integer(R),
14     arbre_binaire(G).
15 arbre_binaire(arb_bin(R, vide, vide)):-
16     integer(R).
17
18 /**
19 * Question 2
20 * dans_arbre_binaire(+E, +B)
21 */
22 dans_arbre_binaire(E, arb_bin(E, G, D)).
23 dans_arbre_binaire(E, arb_bin(R, G, D)):-
24     R \== E,
25     dans_arbre_binaire(E, G);
26     dans_arbre_binaire(E, D).
27
28 /**
29 * Question 3
30 * sous_arbre_binaire(+S, +B)
31 */
32 sous_arbre_binaire(S, S).
33 sous_arbre_binaire(S, arb_bin(E, G, D)):-
34     S \== E,
35     sous_arbre_binaire(S, G);
36     sous_arbre_binaire(S, D).
37
38 /**
39 * Question 4
40 * remplacer(+SA1, +SA2, +B, -B1)
```

```

41 */
42 remplacer(SA1, SA2, SA1, SA2).
43 remplacer(SA1, SA2, vide, vide).
44 remplacer(SA1, SA2, arb_bin(R, G, D), arb_bin(R, G2, D2)):-
45     remplacer(SA1, SA2, G, G2),
46     remplacer(SA1, SA2, D, D2).
47
48 /**
49  * Question 5
50  * isomorphes(+B1, +B2)
51  */
52 isomorphes(A,A).
53 isomorphes(arb_bin(R, G1, D1), arb_bin(R, G2, D2)):-
54     arb_bin(R, G1, D1) \== arb_bin(R, G2, D2),
55     isomorphes(G1, D2),
56     isomorphes(D1, G2).
57 isomorphes(arb_bin(R, G1, D1), arb_bin(R, G2, D2)):-
58     arb_bin(R, G1, D1) \== arb_bin(R, G2, D2),
59     isomorphes(G1, G2),
60     isomorphes(D1, D2).
61
62 /**
63  * Question 6
64  * infixe(+B, -L)
65  * concat(+X, +Y, ?T)
66  */
67 concat([], Y, Y).
68 concat([P|R], Y, [P|T]):-
69     concat(R, Y, T).
70
71 infixe(vide, []).
72 infixe(arb_bin(R, G, D), L):-
73     infixe(G, L1),
74     infixe(D, L2),
75     concat(L1, [R|L2], L).
76
77 /**
78  * Question 7
79  * nb_etiq_sup(+E, +B, ?N)
80  * compter_etiq(+B, ?N)
81  */
82 compter_etiq(vide, 0).
83 compter_etiq(arb_bin(R, G, D), N):-
84     compter_etiq(G, N1),
85     compter_etiq(D, N2),
86     N is N1 + N2 + 1.
87
88 nb_etiq_sup(E, vide, 0).
89 nb_etiq_sup(E, arb_bin(R, G, D), N):-
90     E == R,
91     compter_etiq(D, N).
92 nb_etiq_sup(E, arb_bin(R, G, D), N):-
93     E > R,
94     nb_etiq_sup(E, D, N).
95 nb_etiq_sup(E, arb_bin(R, G, D), N):-
96     E < R,
97     nb_etiq_sup(E, G, N2),
98     compter_etiq(D, N1),

```

```

99  N is N1 + N2 + 1.
100
101 /**
102  * Question 8
103  * insertion_arbre_ordonne(+X, +B1, -B2)
104  * insertion_arbre_ordonne1(+X, +B)
105  */
106 insertion_arbre_ordonne(E, vide, arb_bin(E, vide, vide)).
107 insertion_arbre_ordonne(E, arb_bin(E, G, D), arb_bin(E, G, D)).
108 insertion_arbre_ordonne(E, arb_bin(R, G, D), arb_bin(R, G2, D)):-
109     E < R,
110     insertion_arbre_ordonne(E, G, G2).
111 insertion_arbre_ordonne(E, arb_bin(R, G, D), arb_bin(R, G, D2)):-
112     E > R,
113     insertion_arbre_ordonne(E, D, D2).
114
115 insertion_arbre_ordonne1(E, X):-
116     var(X),
117     X = arb_bin(E, _, _).
118 insertion_arbre_ordonne1(E, arb_bin(E, G, D)).
119 insertion_arbre_ordonne1(E, arb_bin(R, G, D)):-
120     E < R,
121     insertion_arbre_ordonne1(E, G).
122 insertion_arbre_ordonne1(E, arb_bin(R, G, D)):-
123     E > R,
124     insertion_arbre_ordonne1(E, D).

```

2 Tests

Listing 2: tp_arbres_tests.pro

```

1  arbre_binaire(arb_bin(1, arb_bin(2, arb_bin(6, vide, vide), vide),
2      arb_bin(3, arb_bin(4, vide, vide), arb_bin(5, vide, vide)))).
3
4  dans_arbre_binaire(1, arb_bin(1, arb_bin(2, arb_bin(6, vide, vide),
5      vide), arb_bin(3, arb_bin(4, vide, vide), arb_bin(5, vide, vide)))).
6  dans_arbre_binaire(7, arb_bin(1, arb_bin(2, arb_bin(6, vide, vide),
7      vide), arb_bin(3, arb_bin(4, vide, vide), arb_bin(5, vide, vide)))).
8
9  sous_arbre_binaire(arb_bin(3, arb_bin(4, vide, vide), arb_bin(5, vide,
10     vide)), arb_bin(1, arb_bin(2, arb_bin(6, vide, vide), vide),
11     arb_bin(3, arb_bin(4, vide, vide), arb_bin(5, vide, vide)))).
12 sous_arbre_binaire(arb_bin(2, arb_bin(4, vide, vide), arb_bin(5, vide,
13     vide)), arb_bin(1, arb_bin(2, arb_bin(6, vide, vide), vide),
14     arb_bin(3, arb_bin(4, vide, vide), arb_bin(5, vide, vide)))).
15
16 remplacer(arb_bin(3, arb_bin(4, vide, vide), arb_bin(5, vide, vide)),
17     arb_bin(7, vide, vide), arb_bin(1, arb_bin(2, arb_bin(6, vide,
18     vide), vide), arb_bin(3, arb_bin(4, vide, vide), arb_bin(5, vide,
19     vide))), A).
20 remplacer(arb_bin(7, arb_bin(4, vide, vide), arb_bin(5, vide, vide)),
21     arb_bin(7, vide, vide), arb_bin(1, arb_bin(2, arb_bin(6, vide,
22     vide), vide), arb_bin(3, arb_bin(4, vide, vide), arb_bin(5, vide,
23     vide))), A).

```

```

12 isomorphes(arb_bin(3, arb_bin(4, vide, vide), arb_bin(5, arb_bin(6,
    vide, vide), arb_bin(7, vide, vide))), arb_bin(3, arb_bin(5,
    arb_bin(6, vide, vide), arb_bin(7, vide, vide)), arb_bin(4, vide,
    vide))).
13 isomorphes(arb_bin(3, arb_bin(4, vide, vide), arb_bin(5, arb_bin(6,
    vide, vide), arb_bin(7, vide, vide))), arb_bin(3, arb_bin(5,
    arb_bin(7, vide, vide), arb_bin(6, vide, vide)), arb_bin(4, vide,
    vide))).
14 isomorphes(arb_bin(3, arb_bin(4, vide, vide), arb_bin(5, arb_bin(6,
    vide, vide), arb_bin(7, vide, vide))), arb_bin(3, arb_bin(5,
    arb_bin(4, vide, vide), arb_bin(7, vide, vide)), arb_bin(4, vide,
    vide))).
15
16 infixe(arb_bin(1, arb_bin(2, arb_bin(6, vide, vide), vide), arb_bin(3,
    arb_bin(4, vide, vide), arb_bin(5, vide, vide))), L).
17
18 nb_etiq_sup(11, arb_bin(8, arb_bin(4, arb_bin(2, vide, vide),
    arb_bin(6, vide, vide)), arb_bin(12, arb_bin(10, vide, vide),
    vide)), N).
19
20 insertion_arbre_ordonne(7, arb_bin(8, arb_bin(4, arb_bin(2, vide,
    vide), arb_bin(6, vide, vide)), arb_bin(12, arb_bin(10, vide, vide),
    vide)), B2).
21 X = arb_bin(8, arb_bin(4, arb_bin(2, _, _), arb_bin(6, _, _)),
    arb_bin(12, arb_bin(10, _, _), _)), insertion_arbre_ordonne1(7, X).

```