10/08/14 08:56:25 /home-reseau/tchapon/4INFO/Prolog/tp5/tp5_arithmetic_etud.ecl

```
/**
["~/4INFO/Prolog/tp5/tp5_arithmetic_etud.ecl"].
**/
/*
    Question 1.1 : Somme de deux entiers de Peano
*/
add(zero,X,X).
add(s(X),Y,s(S)):- add(X,Y,S).
/*
add(s(zero),zero,S).
    S = s(zero)
    Yes (0.00s cpu)
add(X,Y,s(s(zero))).
    X = zero
    Y = s(s(zero))

    X = s(zero)
    Y = s(zero)

    X = s(s(zero))
    Y = zero
    Yes (0.00s cpu, solution 3)
*/

/*
    Question 1.2 : Différence entre deux entiers de Peano
*/
sub(X,zero,X).
sub(X,X,zero).
sub(s(X),s(Y),S):-sub(X,Y,S).
/*
sub(s(s(zero)),s(zero),S).
    S = s(zero)
    Yes (0.00s cpu, solution 1, maybe more)
*/
/*
    Question 1.3 : Produit de deux entiers de Peano
*/
prod(zero,_,zero).
prod(s(X),Y,S):-prod(X,Y,Z),add(Z,Y,S).
/*
prod(s(s(zero)),s(s(s(zero))),S).
    S = s(s(s(s(s(s(zero))))))
    Yes (0.00s cpu)

prod(zero,s(zero),S).
    S = zero
    Yes (0.00s cpu)

prod(zero,zero,S).
    S = zero
    Yes (0.00s cpu)

prod(s(zero),zero,S).
    S = zero
    Yes (0.00s cpu)
*/
/*
    Question 1.4 : Factorielle d'un entier de Peano
*/
factorial(zero,s(zero)).
factorial(s(X),S):-factorial(X,Y),prod(s(X),Y,S).
/*
factorial(s(s(s((zero)))),S).
    S = s(s(s(s(s(s(zero))))))
    Yes (0.00s cpu)

factorial(s(s(((zero)))),S).
    S = s(s(zero))
    Yes (0.00s cpu)

factorial(s((((zero)))),S).
```

```
 73      S = s(zero)
 74      Yes (0.00s cpu)
 75
 76  factorial(((((zero)))),S).
 77      S = s(zero)
 78      Yes (0.00s cpu)
 79  */
 80  /*
 81      Question 1.5 : Somme de deux entiers en représentation binaire
 82  */
 83  add_bin(X,Y,S):-add_bin_rec(X,Y,0,S).
 84  add_bin_rec([],[],0,[]).
 85  add_bin_rec([],[],1,[1]).
 86  add_bin_rec([],[B|Y],0,[B|Y]).
 87  add_bin_rec([A|X],[],0,[A|X]).
 88  add_bin_rec([],[B|Y],Ret,S):- \==(Ret,0),add_bin_rec([Ret],[B|Y],0,S).
 89  add_bin_rec([A|X],[],Ret,S):- \==(Ret,0),add_bin_rec([A|X],[Ret],0,S).
 90  add_bin_rec([A|X],[B|Y],Ret0,[R|S]):-add_bit(A,B,Ret0,R,Ret),add_bin_rec(X,Y,Ret,S).
 91  /*
 92  add_bin(X,Y,[0,0,1]).
 93      X = []
 94      Y = [0, 0, 1]
 95
 96      X = [0, 0, 1]
 97      Y = []
 98
 99      X = [0]
100      Y = [0, 0, 1]
101
102      X = [0, 0, 1]
103      Y = [0]
104
105      X = [0, 0]
106      Y = [0, 0, 1]
107
108      X = [0, 0, 1]
109      Y = [0, 0]
110
111      X = [0, 1]
112      Y = [0, 1]
113
114      X = [1]
115      Y = [1, 1]
116
117      X = [1, 1]
118      Y = [1]
119
120      X = [1, 0]
121      Y = [1, 1]
122
123      X = [1, 1]
124      Y = [1, 0]
125      Yes (0.00s cpu, solution 11, maybe more)
126
127  add_bin([1,1],[1,0,1,1],S).
128      S = [0, 0, 0, 0, 1]
129      Yes (0.00s cpu, solution 1, maybe more)
130
131  add_bin([1],[0,0,1,1],S).
132      S = [1, 0, 1, 1, 0]
133      Yes (0.00s cpu, solution 1, maybe more)
134
135  add_bin([1,1],[1,1],[0,1,1]).
136      Yes (0.00s cpu, solution 1, maybe more)
137
138  add_bin([0,0,1],Y,[1,1,1]).
139      Y = [1, 1]
140      Yes (0.00s cpu, solution 1, maybe more)
141  */
142
143  /*
144      Question 1.6 : Différence entre deux entiers en représentation binaire
145  */
146  sub_bin(X,Y,S):-add_bin(Y,S,X).
147  /*
```

```
148  sub_bin([1,1],Y,S).
149      Y = []
150      S = [1, 1]
151
152      Y = [1, 1]
153      S = []
154
155      Y = [0]
156      S = [1, 1]
157
158      Y = [0, 1]
159      S = [1]
160
161      Y = [1]
162      S = [0, 1]
163
164      Y = [1, 1]
165      S = [0]
166      Yes (0.00s cpu, solution 6, maybe more)
167
168  sub_bin([1,1,1],[0,0,1],S).
169      S = [1, 1]
170      Yes (0.00s cpu, solution 1, maybe more)
171
172  sub_bin([1,0,1],[0,0,1],[1]).
173      Yes (0.00s cpu, solution 1, maybe more)
174
175  */
176
177  /*
178      Question 1.7 : Produit de deux entiers en représentation binaire
179  */
180  prod_bin([],_,[]).
181  prod_bin([A|X],Y,S):-prod_bin1(A,Y,Res1),prod_bin(X,Y,Res2),add_bin(Res1,[0|Res2],S).
182  prod_bin1(_,[],[]).
183  prod_bin1(A,[B|Y],[R|S]):-prod_bin2(A,B,R),prod_bin1(A,Y,S).
184  prod_bin2(0,_,0).
185  prod_bin2(X,0,0):- \==(X,0).%pour ne pas refaire 0 * 0 = 0
186  prod_bin2(1,1,1).
187  /*
188  prod_bin([1,1],[1,0,1],S).
189      S = [1, 1, 1, 1]
190      Yes (0.00s cpu, solution 1, maybe more)
191
192  prod_bin([1,0,1],[1,0,1],S).
193      S = [1, 0, 0, 1, 1]
194      Yes (0.00s cpu, solution 1, maybe more)
195  */
196  /*
197      Question 1.8 : Factorielle d'un entier en représentation binaire
198  */
199  factorial_bin([],[1]).
200  factorial_bin([0],[1]).
201  factorial_bin(X,F):- sub_bin(X,[1],Sub),factorial_bin(Sub,Fact),prod_bin(X,Fact,F).
202  /*
203  factorial_bin([0,0,1],F).
204      F = [0, 0, 0, 1, 1]
205      Yes (0.00s cpu, solution 1, maybe more)
206
207  factorial_bin(X,[0,1,1]).
208      X = [1, 1]
209      Yes (0.00s cpu, solution 1, maybe more)
210
211  */
212  %%%%%%%%%% Binary representation
213  add_bit(0, 0, 0, 0, 0).
214  add_bit(0, 0, 1, 1, 0).
215  add_bit(0, 1, 0, 1, 0).
216  add_bit(0, 1, 1, 0, 1).
217  add_bit(1, 0, 0, 1, 0).
218  add_bit(1, 0, 1, 0, 1).
219  add_bit(1, 1, 0, 0, 1).
220  add_bit(1, 1, 1, 1, 1).
221
222  %%%%%%%%%% Optional part
```

```
223  evaluate_numbers(N1, M1, N2, M2) :-
224          evaluate(N1, N2),
225          evaluate(M1, M2),
226          number(N2),
227          number(M2).
228
229  evaluate(N, N) :- number(N).
230
231  evaluate(add(N1, M1), N) :-
232          evaluate_numbers(N1, M1, N2, M2),
233          N is N2 + M2.
234
235  evaluate(sub(N1, M1), N) :-
236          evaluate_numbers(N1, M1, N2, M2),
237          N is N2 - M2.
238
239  evaluate(prod(N1, M1), N) :-
240          evaluate_numbers(N1, M1, N2, M2),
241          N is N2 * M2.
242
243  evaluate(eq(N1, M1), Res) :-
244          evaluate_numbers(N1, M1, N2, M2),
245          (
246              N2 = M2, Res = t
247          ;
248              N2 \= M2, Res = f
249          ).
250
251  evaluate(fun(X, Body), fun(X, Body)).
252
253
254  fresh_variables(Expr, Res) :-
255         fresh_variables(Expr, [], Res).
256
257  fresh_variables(X, Assoc, Y) :-
258          var(X),
259          !,
260          assoc(X, Assoc, Y).
261
262  fresh_variables(add(X1, Y1), Assoc, add(X2, Y2)) :-
263          fresh_variables(X1, Assoc, X2),
264          fresh_variables(Y1, Assoc, Y2).
265
266  fresh_variables(prod(X1, Y1), Assoc, prod(X2, Y2)) :-
267          fresh_variables(X1, Assoc, X2),
268          fresh_variables(Y1, Assoc, Y2).
269
270  fresh_variables(sub(X1, Y1), Assoc, sub(X2, Y2)) :-
271          fresh_variables(X1, Assoc, X2),
272          fresh_variables(Y1, Assoc, Y2).
273
274  fresh_variables(eq(X1, Y1), Assoc, eq(X2, Y2)) :-
275          fresh_variables(X1, Assoc, X2),
276          fresh_variables(Y1, Assoc, Y2).
277
278  fresh_variables(if(Cond1, X1, Y1), Assoc, if(Cond2, X2, Y2)) :-
279          fresh_variables(Cond1, Assoc, Cond2),
280          fresh_variables(X1, Assoc, X2),
281          fresh_variables(Y1, Assoc, Y2).
282
283  fresh_variables(Number, _, Number) :- number(Number).
284
285  fresh_variables(fun(X, Body1), Assoc, fun(Y, Body2)) :-
286          fresh_variables(Body1, [(X, Y) | Assoc], Body2).
287
288  fresh_variables(apply(Fun1, Param1), Assoc, apply(Fun2, Param2)) :-
289          fresh_variables(Fun1, Assoc, Fun2),
290          fresh_variables(Param1, Assoc, Param2).
291
292  %Fun = fun(N, fun(F, if(eq(N, 0), 1, prod(N, apply(apply(F, sub(N, 1)), F))))), Factorial = fun(N,
     apply(apply(Fun, N), Fun)), evaluate(apply(Factorial, 42), Res).
```