```
474 Assessed Coursework: Summary for dvn14 of i4
--------------------------------------------------

  Public Tests:
    Part 1 (No-loops):              6 / 6
    Part 1 (Loops):                 6 / 6
    Part 2:                         1 / 1
    Part 3 (Only Attacks):          1 / 4
    Part 3 (Only Supports):         0 / 4
    Part 3 (Attacks and Supports):  1 / 4
    Part 4 (Arguments):             0 / 3
    Part 4 (Attacks):               0 / 2

```

```
%add below the definition of grounded/1
%(i.e. your answer to Part 1)
% do not include any examples

:- dynamic(attacks/2).

attackers(I,A)  :- findall(X,attacks(X,I),A).

defence_from_attacks(_,[]) :- !.
defence_from_attacks(X,[H|T]) :-
        attacks(Y,H),
    \+ attacks(Y,X),
        grounded(Y),
    defence_from_attacks(X,T).

%checks_self_attacks
grounded(X) :-
        attacks(X,X),!,fail.

%checks_attacks_between_themselves
grounded(X) :-
        attacks(X,Y),
        attacks(Y,X),!,fail.

%checks_unattacked
grounded(X) :-
        argument(X),
        \+ attacks(_,X) ,!.

%checks_defence_against_attacks
grounded(X) :-
    argument(X),
        attackers(X,A),
        defence_from_attacks(X,A).
```

```
%add below the definition of strength/2
%(i.e. your answer to Part 3)
% do not include any examples

:- dynamic(supports/2).
:- dynamic(attacks/2).

supporters(I,S) :- findall(X,supports(X,I),S).
attackers(I,A)  :- findall(X,attacks(X,I),A).

base_function(V1,V2,BFV) :-
        BFV is V1+V2-V1*V2.

combination_function(Vo,Vs,Va,C) :-
        Va >= Vs,
        C is Vo-Vo*abs(Vs-Va).

combination_function(Vo,Vs,Va,C) :-
        Va < Vs,
        C is Vo+(1-Vo)*abs(Vs-Va).

strength_function([],V,K) :- !,
        K is V.

strength_function([H|T],0,K) :- !,
        calc_strength(H,Val),
        base_function(0,Val,BFV),
        strength_function(T,BFV,K).

strength_function([H|T],V,K) :-
        calc_strength(H,Val),
        base_function(V,Val,BFV),
        strength_function(T,BFV,K).

calc_strength(I,Val) :-
        \+ attacks(_,I),
        \+ supports(_,I), !,
        base(I,B),
        Val is B.

calc_strength(I,Val) :-
        \+ attacks(_,I),
        supporters(I,S), !,
        strength_function(S,0,Vs),
        base(I,B),
        combination_function(B,Vs,0,C),
        Val is C.

calc_strength(I,Val) :-
        \+ supports(_,I),
        attackers(I,A), !,
        strength_function(A,0,Va),
        base(I,B),
        combination_function(B,0,Va,C),
        Val is C.

calc_strength(I,Val) :-
        supporters(I,S),
    attackers(I,A), !,
        strength_function(A,0,Va),
        strength_function(S,0,Vs),
        base(I,B),
        combination_function(B,Vs,Va,C),
        Val is C.

strength(I,X) :-
        argument(I),
        calc_strength(I,Y),
        X == Y.
```

```
%add below your answer to Part 2

%arguemnts
argument(a0).
argument(a1).
argument(a2).
argument(a3).
argument(a4).
argument(a5).
argument(a6).
argument(a7).
argument(a8).
argument(a9).
argument(a10).
argument(a11).
argument(a12).
argument(a13).
argument(a14).
argument(a15).
argument(a16).
argument(a17).

%attacks
attacks(a1,a0).
attacks(a2,a1).
attacks(a3,a0).
attacks(a3,a2).
attacks(a5,a0).
attacks(a7,a4).
attacks(a7,a6).
attacks(a8,a0).
attacks(a9,a8).
attacks(a13,a0).
attacks(a14,a13).
attacks(a15,a14).
attacks(a17,a3).

%supports
supports(a2,a0).
supports(a2,a4).
supports(a3,a1).
supports(a6,a4).
supports(a8,a1).
supports(a10,a1).
supports(a10,a3).
supports(a11,a10).
supports(a12,a1).
supports(a13,a1).
supports(a15,a13).
supports(a17,a0).
supports(a17,a16).

```

```
% compiling /root/labcat/labcat/engines/lib/prolog/automarker_ft.pl...
%  loading /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/timeout.po...
%  module timeout imported into user
%   loading /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/types.po...
%   module types imported into timeout
%   loaded /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/types.po in module types, 0 msec 4112 bytes
%   loading foreign resource /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/x86_64-linux-glibc2.17/timeout.so in module timeout
%  loaded /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/timeout.po in module timeout, 0 msec 50464 bytes
% compiled /root/labcat/labcat/engines/lib/prolog/automarker_ft.pl in module user, 180 msec 1044720 bytes
SICStus 4.3.5 (x86_64-linux-glibc2.17): Tue Dec  6 10:41:06 PST 2016
Licensed to SP4.3doc.ic.ac.uk
% compiling /tmp/d20180208-36-33rr5r/src/argumentation_autopatch.pl...
%  loading /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/lists.po...
%  module lists imported into user
%   module types imported into lists
%  loaded /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/lists.po in module lists, 10 msec 126544 bytes
% compiled /tmp/d20180208-36-33rr5r/src/argumentation_autopatch.pl in module user, 10 msec 150416 bytes
yes
% compiling /tmp/d20180208-36-33rr5r/src/grounded.pl...
% compiled /tmp/d20180208-36-33rr5r/src/grounded.pl in module submittedGrounded, 0 msec 2816 bytes
yes
% compiling /tmp/d20180208-36-33rr5r/src/df_quad.pl...
% compiled /tmp/d20180208-36-33rr5r/src/df_quad.pl in module submittedDFQUAD, 0 msec 10656 bytes
yes
% compiling /tmp/d20180208-36-33rr5r/src/angrymen.pl...
% compiled /tmp/d20180208-36-33rr5r/src/angrymen.pl in module submittedANGRY, 0 msec 9152 bytes
yes
% compiling /tmp/d20180208-36-33rr5r/src/aba.pl...
% compiled /tmp/d20180208-36-33rr5r/src/aba.pl in module submittedABA, 0 msec 0 bytes
yes
% compiling /tmp/d20180208-36-33rr5r/src/ma_grounded.pl...
% compiled /tmp/d20180208-36-33rr5r/src/ma_grounded.pl in module modelGrounded, 10 msec 4432 bytes
yes
% compiling /tmp/d20180208-36-33rr5r/src/ma_df_quad.pl...
% compiled /tmp/d20180208-36-33rr5r/src/ma_df_quad.pl in module modelDFQUAD, 0 msec 6256 bytes
yes
% compiling /tmp/d20180208-36-33rr5r/src/ma_angrymen.pl...
% compiled /tmp/d20180208-36-33rr5r/src/ma_angrymen.pl in module modelANGRY, 0 msec 7056 bytes
yes
% compiling /tmp/d20180208-36-33rr5r/src/ma_aba.pl...
%  loading /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/sets.po...
%  module sets imported into modelABA
%   module lists imported into sets
%  loaded /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/sets.po in module sets, 0 msec 23456 bytes
% compiled /tmp/d20180208-36-33rr5r/src/ma_aba.pl in module modelABA, 0 msec 28112 bytes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
```

```
120: yes
121: yes
122: yes
123: yes
124: yes
125: yes
126: yes
127: yes
128: yes
129: yes
130: yes
131: yes
132: yes
133: yes
134: yes
135: yes
136: yes
137: yes
138: yes
139: yes
140: yes
141: yes
142: yes
143: yes
144: yes
145: yes
146: yes
147: yes
148: yes
149: yes
150: yes
151: yes
152: yes
153: yes
154: yes
155: yes
156: yes
157: yes
158: yes
159: yes
160: yes
161: yes
162: yes
163: yes
164: yes
165: yes
166: yes
167: yes
168: yes
169: yes
170: yes
171: yes
172: yes
173: yes
174: yes
175: yes
176: yes
177: yes
178: yes
179: yes
180: yes
181: yes
182: yes
183: yes
184: yes
185: yes
186: yes
187: yes
188: yes
189: yes
190: yes
191: yes
192: yes
193: yes
194: yes
195: yes
196: yes
197: yes
198: yes
199: yes
200: yes
201: yes
202: yes
203: yes
204: yes
205: yes
206: yes
207: yes
208: yes
209: yes
210: yes
211: yes
212: yes
213: yes
214: yes
215: yes
216: yes
217: yes
218: yes
219: yes
220: yes
221: yes
222: yes
223: yes
224: yes
225: yes
226: yes
227: yes
228: yes
229: yes
230: yes
231: yes
232: yes
233: yes
234: yes
235: yes
236: yes
237: yes
238: yes
239: yes
240: yes
241: yes
242: yes
243: yes
244: yes
245: yes
246: yes
247: yes
248: yes
249: yes
250: yes
251: yes
252: yes
253: yes
```

```
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
yes
! Existence error in submittedABA:argument/1
! procedure submittedABA:argument/1 does not exist
! goal:  submittedABA:argument((a,_383))
! Existence error in submittedABA:argument/1
! procedure submittedABA:argument/1 does not exist
! goal:  submittedABA:argument((y,_383))
! Existence error in submittedABA:argument/1
! procedure submittedABA:argument/1 does not exist
! goal:  submittedABA:argument((z,_383))
yes
! Existence error in submittedABA:attacks/2
! procedure submittedABA:attacks/2 does not exist
! goal:  submittedABA:attacks((_383,_385),(x,_391))
! Existence error in submittedABA:attacks/2
! procedure submittedABA:attacks/2 does not exist
! goal:  submittedABA:attacks((y,_385),(y,_391))
yes
```

```
===============================================================================
        474 Arg&MAS::Assessed Coursework
        Submission: dvn14
===============================================================================

===============================================================================
Part 1 (No-loops)
---------- Test 1 ::a not in GE -----------------------------------------------

| ? grounded(a).
no         %% correct

---------- Test 2 ::b not in GE -----------------------------------------------

| ? grounded(b).
no         %% correct

---------- Test 3 ::c in GE ---------------------------------------------------

| ? grounded(c).
yes        %% correct

---------- Test 4 ::d not in GE -----------------------------------------------

| ? grounded(d).
no         %% correct

---------- Test 5 ::e in GE ---------------------------------------------------

| ? grounded(e).
yes        %% correct

---------- Test 6 ::f in GE ---------------------------------------------------

| ? grounded(f).
yes        %% correct


===============================================================================
Part 1 (No-loops)
TESTS PASSED:  6 / 6 ;  MARKS:  6 / 6
===============================================================================

===============================================================================
Part 1 (Loops)
---------- Test 1 ::a not in GE -----------------------------------------------

| ? grounded(a).
no         %% correct

---------- Test 2 ::b not in GE -----------------------------------------------

| ? grounded(b).
no         %% correct

---------- Test 3 ::c not in GE -----------------------------------------------

| ? grounded(c).
no         %% correct

---------- Test 4 ::d not in GE -----------------------------------------------

| ? grounded(d).
no         %% correct

---------- Test 5 ::e not in GE -----------------------------------------------

| ? grounded(e).
no         %% correct

---------- Test 6 ::f in GE ---------------------------------------------------

| ? grounded(f).
yes        %% correct


===============================================================================
Part 1 (Loops)
TESTS PASSED:  6 / 6 ;  MARKS:  6 / 6
===============================================================================

===============================================================================
Part 2
---------- Test 1 ::correct set of arguments considered -----------------------

| ? arguments_differences.

These arguments should not be mined:

These arguments are missing:

yes
These arguments should not be mined:

These arguments are missing:

        %% correct


===============================================================================
Part 2
TESTS PASSED:  1 / 1 ;  MARKS:  1 / 1
===============================================================================

===============================================================================
Part 3 (Only Attacks)
---------- Test 1 ::strength of d ---------------------------------------------

| ? strength(d,_947).
no         %% WRONG

---------- Test 2 ::strength of c ---------------------------------------------

| ? strength(c,_947).
no         %% WRONG

---------- Test 3 ::strength of b ---------------------------------------------

| ? strength(b,_947).
no         %% WRONG

---------- Test 4 ::does a have strength 0.234375? ----------------------------

| ? strength(a,0.234375).
yes        %% correct


===============================================================================
Part 3 (Only Attacks)
TESTS PASSED:  1 / 4 ;  MARKS:  1 / 4
===============================================================================

===============================================================================
Part 3 (Only Supports)
```

```
---------- Test 1 ::strength of d -----------------------------------------------

| ? strength(d,_947).
no         %% WRONG

---------- Test 2 ::strength of c -----------------------------------------------

| ? strength(c,_947).
no         %% WRONG

---------- Test 3 ::strength of b -----------------------------------------------

| ? strength(b,_947).
no         %% WRONG

---------- Test 4 ::strength of a -----------------------------------------------

| ? strength(a,_947).
no         %% WRONG


================================================================================
Part 3 (Only Supports)
TESTS PASSED:  0 / 4 ;  MARKS:  0 / 4
================================================================================

================================================================================
Part 3 (Attacks and Supports)
---------- Test 1 ::strength of d -----------------------------------------------

| ? strength(d,_947).
no         %% WRONG

---------- Test 2 ::strength of c -----------------------------------------------

| ? strength(c,_947).
no         %% WRONG

---------- Test 3 ::strength of b -----------------------------------------------

| ? strength(b,_947).
no         %% WRONG

---------- Test 4 ::strength of a is 0.390625? ----------------------------------

| ? strength(a,0.390625).
yes        %% correct


================================================================================
Part 3 (Attacks and Supports)
TESTS PASSED:  1 / 4 ;  MARKS:  1 / 4
================================================================================

================================================================================
Part 4 (Arguments)
---------- Test 1 ::argument for single assumptions ----------------------------

| ? argument((a,[a])).
Test 1:  FATAL ERROR!!

---------- Test 2 ::argument for non-assumptions -------------------------------

| ? argument((y,[b])).
Test 2:  FATAL ERROR!!

---------- Test 3 ::non- arguments ---------------------------------------------

| ? argument((z,_957)).

Test 3:  FATAL ERROR!!

================================================================================
Part 4 (Arguments)
TESTS PASSED:  0 / 3   (Fatal errors: 3);  MARKS:  0 / 3
================================================================================

================================================================================
Part 4 (Attacks)
---------- Test 1 ::no attacks against any arguments with claim x --------------

| ? attacks(_923,(x,_989)).
Test 1:  FATAL ERROR!!

---------- Test 2 ::self-attacking arguments -----------------------------------

| ? attacks((y,[b]),(y,[b])).

Test 2:  FATAL ERROR!!

================================================================================
Part 4 (Attacks)
TESTS PASSED:  0 / 2   (Fatal errors: 5);  MARKS:  0 / 2
================================================================================


==================================== SUMMARY ===================================

TESTS PASSED (dvn14):  15 / 30   (Fatal errors: 8);  MARKS:  15 / 30

================================================================================
```