

NOEH, Noah-Vincenz (nn4718)



531 tk106 1
a5 nn4718 v2



Electronic submission



Mon - 22 Oct 2018
18:40:55

nn4718

Exercise Information

Module: 531 Prolog
Exercise: 1 (LAB)
Title: Formulas
FAO: Kimber, Timothy (tk106)

Issued: Wed - 10 Oct 2018
Due: Mon - 22 Oct 2018
Assessment: Individual
Submission: Electronic

Student Declaration - Version 2

- I acknowledge the following people for help through our original discussions:

Noah-Vincenz Noeh (nn4718)

Signed: (electronic signature)

Date: 2018-10-20 16:22:13

For Markers only: (circle appropriate grade)

NOEH, (nn4718)	Noah-Vincenz	01562775	a5	2018-10-20 16:22:13	A*	A	B	C	D	E	F
-------------------	--------------	----------	----	---------------------	----	---	---	---	---	---	---

Prolog Formulas/TestSummary

```
1: Prolog Formulas: Summary for nn4718 of a5
2: -----
3:
4:   Full Tests:
5:     Task 1 (wff/1):      5 / 6
6:     Task 2 (cls/1):      5 / 5
7:     Task 3 (ats/2):      4 / 4
8:     Task 4 (t_value/3): 10 / 10
9:
```

TestSummary.txt: 1/1

Noah-Vincenz Noeh - nn4718: a5

```

1: % 531 Prolog
2: % Assessed Exercise 1
3: % formulas.pl
4:
5:
6: % Write your answers to the exercise here
7: :- consult(support).
8:
9: % Task 1: wff(+F) %+F means ground -F means variable
10: % wff(F) holds when F is a (well-formed) formula.
11:
12: % If formula A is not ground we want to fail without further backtracking.
13: wff(A) :-
14:     \+ ground(A), !, fail.
15:
16: wff(A) :-
17:     logical_atom(A).
18:
19: wff(neg(A)) :-
20:     logical_atom(A).
21:
22: wff(and(A, B)) :-
23:     wff(A),
24:     wff(B).
25:
26: wff(or(A, B)) :-
27:     wff(A),
28:     wff(B).
29:
30: wff(imp(A, B)) :-
31:     wff(A),
32:     wff(B).
33:
34:
35: % Task 2: cls(+F)
36: % cls(F) holds when the formula F is a clause; a clause is either a literal or
37: % a disjunction of literals, and a literal is either an atom or a negated atom.
38:
39: % If formula A is not ground we want to fail without further backtracking.
40: cls(A) :-
41:     \+ ground(A), !, fail.
42:
43: cls(A) :-
44:     logical_atom(A).
45:
46: cls(neg(A)) :-
47:     logical_atom(A).
48:
49: cls(or(A, B)) :-
50:     cls(A),
51:     cls(B).
52:
53:
54: % Task 3: ats(+F, -As)
55: % given the formula F, returns As as a duplicate-free list (in any order) of
56: % the atoms in F.
57:
58: % If formula A is not ground we want to fail without further backtracking.
59: ats(F, _) :-
60:     \+ ground(F), !, fail.
61:
62: % If it is ground then we want to call createList, which recursively creates an in
63: % itially empty accumulator and uses this to add each logical atom as a Head element to As
64: ats(F, As) :-
65:     createList(F, [], As), !.
66:
67: createList(A, Acc, As2),
68: createList(B, As2, As).
69:
70: createList(or(A, B), Acc, As) :-
71:     createList(A, Acc, As2),
72:     createList(B, As2, As).
73:
74: createList(neg(A), Acc, As) :-
75:     createList(A, Acc, As).
76:
77: createList(imp(A, B), Acc, As) :-
78:     createList(A, Acc, As2),
79:     createList(B, As2, As).
80:
81: createList(A, Acc, As) :-
82:     logical_atom(A),
83:     (\+ member(A, Acc) -> singleAtom([A|Acc], As); % We only want to add the
atom to the list if it is not already contained within the accumulator
84:     singleAtom(Acc, As)).
85:
86: singleAtom(As, As).
87:
88:
89: % Task 4: t_value(+F, +Val, -V)
90: % Calculates the truth value V of the formula F, given the valuation Val.
91:
92: t_value(F, Val, _) :-
93:     \+ ground(Val), !, fail; % checks if Val is ground
94:
95:     \+ (ats(F, ListOfAtomsInF), % checks if F is a ground wff of logical ato
ms and stores a list of the atoms present in F
96:     atomsOf(Val, ListOfAtomsInF)), !, fail.
97:
98: t_value(F, Val, V) :-
99:     ( t1_value(F, Val, V) ) -> assignTrue(V); % if the truth value of F is t
rue then let V be true; else false
100:     assignFalse(V).
101:
102: % Recursively checks if Val ([Head|Tail]) only contains logical_atoms occuring in
F
103: atomsOf([Head|Tail], ListOfAtomsInF) :-
104:     (logical_atom(Head), member(Head, ListOfAtomsInF)) -> atomsOf(Tail, ListOfAtomsI
nF).
105:
106: atomsOf([], _).
107:
108: % t1_value function computes the truth value of a formula F - it also checks if F
is a wff
109: t1_value(A, Val, _) :-
110:     member(A, Val).
111:
112: t1_value(neg(A), Val, V) :-
113:     \+ t1_value(A, Val, V).
114:
115: t1_value(and(A,B), Val, _) :-
116:     t1_value(A, Val, _V1),
117:     t1_value(B, Val, _V2).
118:
119: t1_value(or(A,B), Val, _) :-
120:     t1_value(A, Val, _V1);
121:     t1_value(B, Val, _V2).
122:
123: t1_value(imp(A,B), Val, _) :-
124:     \+ t1_value(A, Val, _V1);
125:     t1_value(B, Val, _V2).
126:
127: assignTrue(t).

```

```
128: assignFalse(f).
```

```
1: % compiling /root/labcat/labcat/engines/lib/prolog/automarker.pl...
2: % loading /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/timeout.po...
3: % module timeout imported into user
4: % loading /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/types.po...
5: % module types imported into timeout
6: % 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
7: % 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
8: % loaded /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/timeout.po in module timeout, 10 msec 52576 bytes
9: % compiled /root/labcat/labcat/engines/lib/prolog/automarker.pl in module user, 22 0 msec 1055536 bytes
10: SICStus 4.3.5 (x86_64-linux-glibc2.17): Tue Dec 6 10:41:06 PST 2016
11: Licensed to SP4.3doc.ic.ac.uk
12: % compiling /tmp/d20181022-41-vtyfp/src/solution.pl...
13: % loading /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/lists.po...
14: % module lists imported into autom
15: % module types imported into lists
16: % loaded /usr/lib/sicstus4.3.5/bin/sp-4.3.5/sicstus-4.3.5/library/lists.po in module lists, 10 msec 127696 bytes
17: % compiled /tmp/d20181022-41-vtyfp/src/solution.pl in module autom, 10 msec 148288 bytes
18: yes
19: % compiling /tmp/d20181022-41-vtyfp/src/formulas.pl...
20: % consulting /tmp/d20181022-41-vtyfp/src/support.pl...
21: % consulted /tmp/d20181022-41-vtyfp/src/support.pl in module submission, 0 msec 5376 bytes
22: % compiled /tmp/d20181022-41-vtyfp/src/formulas.pl in module submission, 0 msec 18880 bytes
23: yes
24: yes
25: yes
26: yes
27: yes
28: yes
29: yes
30: yes
31: yes
32: yes
33: yes
34: yes
35: yes
36: yes
37: yes
38: yes
39: yes
40: yes
41: As = [q,p] ?
42: yes
43: As = [p,q] ?
44: yes
45: yes
46: yes
47: yes
48: yes
49: yes
50: yes
51: yes
52: yes
53: yes
54: yes
55: yes
56: yes
57: yes
58: yes
```

```

1: =====
2:      531 Prolog: Exercise 1 (logical formulas)
3:      Submission: nn4718
4: =====
5:
6: =====
7: Task 1 (wff/1)
8:
9: ----- Test 1 :: wff:: valid atom -----
10:
11: | ? wff(pedro).
12: yes      %% correct
13:
14: ----- Test 2 :: wff:: valid negation -----
15:
16: | ? wff(neg(p)).
17: yes      %% correct
18:
19: ----- Test 3 :: wff:: invalid atom -----
20:
21: | ? wff(neg(**)).
22: no       %% correct
23:
24: ----- Test 4 :: wff:: valid compound formula -----
25:
26: | ? wff(neg(imp(and(p,q),p))).
27: no       %% WRONG
28:
29: ----- Test 5 :: wff:: invalid connective -----
30:
31: | ? wff(neg(iff(and(p,q),p))).
32: no       %% correct
33:
34: ----- Test 6 :: wff:: non-ground formula -----
35:
36: | ? wff(imp(imp(_971,q),p)).
37: no       %% correct
38:
39:
40: =====
41: Task 1 (wff/1)
42: TESTS PASSED: 5 / 6
43: =====
44:
45: =====
46: Task 2 (cls/1)
47:
48: ----- Test 1 :: cls:: valid literal -----
49:
50: | ? cls(p).
51: yes      %% correct
52:
53: ----- Test 2 :: cls:: valid literal -----
54:
55: | ? cls(neg(p)).
56: yes      %% correct
57:
58: ----- Test 3 :: cls:: valid disjunction -----
59:
60: | ? cls(or(neg(p),or(q,neg(r)))).
61: yes      %% correct
62:
63: ----- Test 4 :: cls:: invalid negated formula -----
64:
65: | ? cls(neg(or(neg(p),q))).
66: no       %% correct
67:

```

```

68: ----- Test 5 :: cls:: non-ground formula -----
69:
70: | ? cls(or(neg(p),or(_1029,neg(r)))).
71: no       %% correct
72:
73:
74: =====
75: Task 2 (cls/1)
76: TESTS PASSED: 5 / 5
77: =====
78:
79: =====
80: Task 3 (ats/2)
81:
82: ----- Test 1 :: ats:: wff contains p,q -----
83:
84: | ? find As: ats(imp(imp(p,q),p),As).
85: As = [q,p] ;      %% correct
86:
87: ----- Test 2 :: ats:: wff contains p,q -----
88:
89: | ? find As: ats(and(q,imp(p,q)),As).
90: As = [p,q] ;      %% correct
91:
92: ----- Test 3 :: ats:: not a wff -----
93:
94: | ? find As: ats(and(q,iff(p,q)),As).
95: No solution      %% correct
96:
97: ----- Test 4 :: ats:: non-ground formula -----
98:
99: | ? find As: ats(and(q,imp(_1063,q)),As).
100: No solution      %% correct
101:
102:
103: =====
104: Task 3 (ats/2)
105: TESTS PASSED: 4 / 4
106: =====
107:
108: =====
109: Task 4 (t_value/3)
110:
111: ----- Test 1 :: t_value:: Valuation atom not in formula -----
112:
113: | ? find V: t_value(or(and(p,q),and(neg(p),neg(q))),[p,q,r],V).
114: No solution      %% correct
115:
116: ----- Test 2 :: t_value:: formula is true -----
117:
118: | ? find V: t_value(or(and(p,q),and(neg(p),neg(q))),[p,q],V).
119: V = t ;
120: No more solutions (All correct)
121: No missing solutions
122:
123: ----- Test 3 :: t_value:: invalid connective -----
124:
125: | ? find V: t_value(or(iff(p,q),and(neg(p),neg(q))),[p,q],V).
126: No solution      %% correct
127:
128: ----- Test 4 :: t_value:: formula is true -----
129:
130: | ? find V: t_value(imp(r,or(and(p,q),and(neg(p),neg(q)))),[p,q],V).
131: V = t ;
132: No more solutions (All correct)
133: No missing solutions
134:

```

```
135: ----- Test 5 :: t_value:: non-ground valuation -----
136:
137: | ? find V: t_value(imp(r,or(and(p,q),and(neg(p),neg(q)))),[p,_1345],V).
138: No solution      %% correct
139:
140: ----- Test 6 :: t_value:: non atom in valuation -----
141:
142: | ? find V: t_value(imp(r,or(and(p,q),and(neg(p),neg(q)))),[p,[q]],V).
143: No solution      %% correct
144:
145: ----- Test 7 :: t_value:: non atom in valuation -----
146:
147: | ? find V: t_value(imp(r,or(and(p,q),and(neg(p),neg(q)))),[p,**],V).
148: No solution      %% correct
149:
150: ----- Test 8 :: t_value:: formula is false -----
151:
152: | ? find V: t_value(imp(r,or(and(p,q),and(neg(p),neg(q)))),[p,r],V).
153: V = f ;
154: No more solutions (All correct)
155: No missing solutions
156:
157: ----- Test 9 :: t_value:: formula is true -----
158:
159: | ? find V: t_value(imp(r,or(and(p,q),and(neg(p),neg(q)))),[r],V).
160: V = t ;
161: No more solutions (All correct)
162: No missing solutions
163:
164: ----- Test 10 :: t_value:: formula is true -----
165:
166: | ? find V: t_value(imp(r,or(and(p,q),and(neg(p),neg(q)))),[],V).
167: V = t ;
168: No more solutions (All correct)
169: No missing solutions
170:
171:
172: =====
173: Task 4 (t_value/3)
174: TESTS PASSED: 10 / 10
175: =====
176:
177:
178: ===== SUMMARY (nn4718) =====
179:
180: TESTS PASSED: 24 / 25
181: =====
```