

lib/main/group_unittest.ath

```

1  ## Test group concepts and theorems
2
3  load "group"
4
5  module Test {
6
7  open Group
8
9  define Group0 := no-renaming
10
11 assert (Group0 (theory-axioms 'Group))
12
13 #####
14 #
15 #(set-debug-mode "rewriting")
16
17 (test-proofs [Group.left-inverse Group.double-negation
18              Group.unique-negation Group.neg-plus] 'Group)
19
20 #####
21 ## Create some dummy symbols to plug into Group theory
22
23 declare plus: (T) [T T] -> T
24
25 declare negs: (T) [T] -> T
26
27 declare zeros: (T) [] -> T
28
29 # Show that the proofs work with this different set of symbols.
30
31 define Group1 := (renaming |{Group.+ := plus, Group.U- := negs, Group.<0> := zeros}|)
32
33 assert (Group1 (theory-axioms 'Group))
34
35 # This time, don't prove 'Left-Inverse before testing 'Double-Negation;
36 # it should thus be proved on the fly.
37
38 (!prove-property double-negation Group1 Group.theory)
39
40 (!prove-property unique-negation Group1 Group.theory)
41
42 (!prove-property neg-plus Group1 Group.theory)
43
44 # Although it was proved during the proof of Double-Negation, it wasn't
45 # left in the assumption base, so !property has to reprove it.
46 (!prove-property left-inverse Group1 Group.theory)
47
48 #####
49
50 declare plus': (T) [T T] -> T
51
52 declare neg': (T) [T] -> T
53
54 declare zero': (T) [] -> T
55
56 declare minus': (T) [T T] -> T
57
58 define Abelian-Group0 := (renaming |{Group.+ := plus', Group.U- := neg',
59                                   Group.<0> := zero', Group.- := minus'}|)
60
61 assert (Abelian-Group0 (theory-axioms Abelian-Group.theory))
62
63 (!prove-property left-inverse Abelian-Group0 Abelian-Group.theory)
64
65 (!prove-property double-negation Abelian-Group0 Abelian-Group.theory)
66
67 (!prove-property unique-negation Abelian-Group0 Abelian-Group.theory)

```

```
68
69 (!prove-property neg-plus Abelian-Group0 Abelian-Group.theory)
70
71 (!prove-property Abelian-Group.neg-plus Abelian-Group0 Abelian-Group.theory)
72
73 #####
74
75 define MG0 := no-renaming
76
77 assert (MG0 (theory-axioms MG.theory))
78
79 (!prove-property left-inverse MG0 MG.theory)
80
81 (!prove-property double-negation MG0 MG.theory)
82
83 (!prove-property unique-negation MG0 MG.theory)
84
85 (!prove-property neg-plus MG0 MG.theory)
86
87 #####
88
89 declare times1: (T) [T T] -> T
90
91 declare one1: (T) [] -> T
92
93 declare inv1: (T) [T] -> T
94
95 declare div1: (T) [T T] -> T
96
97 define MG1 :=
98   (renaming |{MSG.* := times1, MM.<1> := one1, MG.inv := inv1, MG./ := div1}|)
99
100 assert (MG1 (theory-axioms MG.theory))
101
102 (!prove-property left-inverse MG1 MG.theory)
103
104 (!prove-property double-negation MG1 MG.theory)
105
106 (!prove-property unique-negation MG1 MG.theory)
107
108 (!prove-property neg-plus MG1 MG.theory)
109
110 } # Test
```