lib/main/group_unittest.ath

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

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