

lib/main/integer-integral-domain.ath

```

1 load "integral-domain.ath"
2 load "integer-times.ath"
3
4 open Z
5
6 define Integer-Ring :=
7   (renaming [Commutative-Ring.+ + Commutative-Ring.* *
8             Commutative-Ring.<0> zero Commutative-Ring.U- negate
9             Commutative-Ring.- -])
10
11 (print-instance-check Integer-Ring Commutative-Ring.Theory)
12
13 define Integer-Ring-1 :=
14   (renaming [Commutative-Ring.+ + Commutative-Ring.* *
15             x`Commutative-Ring.<0> zero Commutative-Ring.U- negate
16             Commutative-Ring.- -
17             Commutative-Ring-With-Identity.<1> one])
18
19 (print-instance-check Integer-Ring-1 Commutative-Ring-With-Identity.Theory)
20
21 assert (theory-axioms Commutative-Ring-With-Identity.Theory)
22
23 (!property Group.Double-Negation no-renaming Group.Theory)
24
25 (!by-instance-check Group.Double-Negation Integer-Ring-1 Commutative-Ring-With-Identity.Theory)
26
27 (!by-instance-check Group.Left-Inverse Integer-Ring-1 Commutative-Ring-With-Identity.Theory)
28
29 (!property Group.Left-Inverse no-renaming Commutative-Ring-With-Identity.Theory)
30
31 (!property Group.Left-Inverse Integer-Ring-1 Commutative-Ring-With-Identity.Theory)
32
33 (!property Group.Unique-Negation no-renaming Group.Theory)
34
35 (!by-instance-check Group.Unique-Negation Integer-Ring-1 Commutative-Ring-With-Identity.Theory)
36
37 ##
38
39 define ZID :=
40   (renaming [Commutative-Ring.+ + Commutative-Ring.* *
41             Commutative-Ring.<0> zero Commutative-Ring.U- negate
42             Commutative-Ring.- - Commutative-Ring-With-Identity.<1> one])
43
44 (print-instance-check ZID Integral-Domain.Theory)

```