

Chapter 1

Code Sample

```
abstract sig Step { next : lone Step }
abstract sig StopStep extends Step { }{ next = none }
abstract sig StartStep extends Step { }

abstract sig Op { start : one StartStep }

sig Start1 extends StartStep { }
fact { Start1.next = Stop1 }
sig Stop1 extends StopStep { }{ }
sig Op1 extends Op { }
fact { Op1.start = Start1 }

sig Start2_A extends StartStep { }
fact { Start2_A.next = Start2_B }
sig Stop2 extends StopStep {}{}
//sig Start2_B extends StartStep { }{ next = Start2_A }
sig Start2_B extends StartStep { }
fact { Start2_B.next = Stop2 }
sig Op2 extends Op { "a" }
fact { Op2.start = Start2_A }

assert test {
  // no o : Op | (all f : StopStep | o.start.^next != f)
  all o : Op | (some f : StopStep | o.start.^next = f)
}
check test for 3

assert test2 {
  no p : Page | p in p.^redirectsTo
}
check test2 for 1
```

```
assert b {  
  all a : A | no b : B | b in a  
}  
check b for 3
```

Bibliography