Table of Contents

MoonBit QuickCheck Package

MoonBit QuickCheck package provides property-based testing capabilities by generating random test inputs.

Basic Usage

Generate random values of any type that implements the Arbitrary trait:

```

test "basic generation" {
  let b : Bool = @quickcheck gen()
  inspect(b, content="true")
  let x : Int = @quickcheck gen()
  inspect(x, content="0")


  let sized : Array[Int] = @quickcheck gen(size=5)
  inspect(sized length() <= 5, content="true")
}
```

Multiple Samples

Generate multiple test cases using the samples function:

```

test "multiple samples" {
  let ints : Array[Int] = @quickcheck samples(5)
  inspect(ints, content="[0, 0, 0, -1, -1]")
  let strings : Array[String] = @quickcheck samples(12)
  inspect(
    strings[5:10],
    content=(
      #|["E\b\u{0f} ", "", "K\u{1f}[", "!@", "xvLxb"]
    ),
  )
}
```

Built-in Types

QuickCheck provides Arbitrary implementations for all basic MoonBit types:

```

test "builtin types" {

  let v : (Bool, Char, Byte) = @quickcheck gen()
  inspect(v, content="(true, '#', b'\\x12')")

  let v : (Int, Int64, UInt, UInt64, Float, Double, BigInt) = @quickchec
  inspect(v, content="(0, 0, 0, 0, 0.1430625319480896, 0.330984466952546


  let v : (String, Bytes, Iter[Int]) = @quickcheck gen()
  inspect(
    v,
    content=(
      #|("", b"", [])
    ),
  )
}
```

Custom Types

Implement Arbitrary trait for custom types:

```

struct Point {
  x : Int
  y : Int
} derive(Show)


impl Arbitrary for Point with arbitrary(size, r0) {
  let r1 = r0 split()
  let y = @quickcheck Arbitrary::arbitrary(size, r1)
  { x: @quickcheck Arbitrary::arbitrary(size, r0), y }
}


test "custom type generation" {
  let point : Point = @quickcheck gen()
  inspect(point, content="{x: 0, y: 0}")
  let points : Array[Point] = @quickcheck samples(10)
  inspect(
    points[6:],
    content="[{x: 0, y: 1}, {x: -1, y: -5}, {x: -6, y: -6}, {x: -1, y: 7
  )
}
```

The package is useful for writing property tests that verify code behavior acros
s a wide range of randomly generated inputs.