Everything on Model-Checking

First Order Logic

Alloy is based on first order logic. Every specification contains some undefined elements these form the signature of the specification

A structure of a signature is an assignment of specific values to each of the variables in the signature

sig Student { }	There are students.
one sig JaneDoe **extends** Student { }	There is a student called Jane Doe.

sig Module {
  class : **set** Student
}
	There are modules.
	There is a relation, class, between Module and Student.

one sig CSC9P6 **extends** Module { }	There is a module called CSC9P6.
fact { JaneDoe in CSC9P6.class }	Jane Doe is in the CSC9P6 class.

```
Student =  {JaneDoe, Student0, Student1}
JaneDoe = {JaneDoe}
Module =  {CSC9P6, Module0}
Class =  {CSC9P6 -> Student0, CSC9P6 -> JaneDoe}
CSC9P6 =  {CSC9P6}
```

```
Student =  {JaneDoe, Student0, Student1}
JaneDoe = {JaneDoe}
Module =  {CSC9P6, Module0}
Class =  {CSC9P6 -> Student0, CSC9P6 -> Student1}
CSC9P6 =  {CSC9P6}
```

A model of a specification is a structure for the specification which makes all the statements in the specification true.

A logical statement is consistent if there exists at least one model of that statement. There's at least one possible world in which this statement is true

A logical statement is valid if every structure is a model of that statement. The statement is true in all possible worlds.

Run a predicate → check if it is consistent
Check an assertion → check if assertion is valid

Run a predicate to find out if something can possible be true. Check an assertion to find out if something must always be true

Alloy only checks models up to a certain size → scope. The default scope is 3. This can be modified

Limitations of model checking

Because of the use of a finite scope, model-checking in alloy has some limitations

1) When a predicate is run, if a model instance is found we can be certain that the predicate is consistent, but if no instance is found, we cannot conclude that the predicate is inconsistent

   It may be inconsistent or a model instance could be found if a larger scope was used

2) When an assertion is checked, if a counter-example is found, we can be certain the assertion is invalid, but if no counter-example is found, we cannot be sure that the assertion is valid.

   It may be valid or maybe a counter-example can be found by using a larger scope

Small Scope Hypothesis

Despite the limitation, it can still be argued that alloy is a useful tool

The basic statement to support that is the small scope hypothesis:

Most bugs have small counter-examples. If an assertion is invalid, it possible has a small counter-example. If you examine all small cases, you're likely to find a counter-example