

Sets Types and Basic Notation

Mairead Meagher

Waterford Institute of Technology,

2017



Waterford Institute *of* Technology

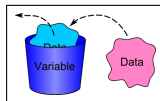
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

- Sets and Types
- Set Operations
- Set Comprehension

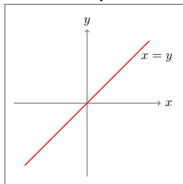
- Z Specification use mathematics to express the properties of systems.
- Z is a mathematical language based on standard set notation.
- Sets in Z
 - A set is a collection of values called elements or members.
 - In Z, all possible values of a set are considered to have something in common and are said to have the same type.
 - Any set is considered to be a subset of its type.

- Need to distinguish between mathematical variable (Z) and programming variable:
 - Programming variable - name of a location in some store.

Interested in its value, can change it.



- Mathematical variable (used in Z) - a value that has a type, the value is often non-specific and has a range of values, and



can be related

to other values (e.g. $x = y$)

- Z adopts a simple approach to typing.
- A type denotes a non-empty set of values which is treated as **maximal**
 - (maximal means that suppose that S is a set of type τ , all values of type τ are contained in set S).
- Moreover, S contains nothing but τ values. This notion of type has two important aspects:

- a type denotes a set that is not contained in any wider set (i.e. subtypes do not exist in Z)
- the type of any Z expression can be determined by an algorithm, irrespective of the value that the expression denotes (type checking is thus possible)

- Every Z specification needs one or more basic types with which to describe objects whose internal make-up is of no relevance to the specification. Values of basic types are regarded as atomic.
- Types can be one of:
 - Built-in types
 - Free-types
 - Basic types