Sets Types and Basic Notation

Mairead Meagher

Waterford Institute of Technology,

2017

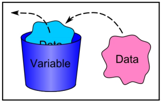
Sets Types and Basic Notation

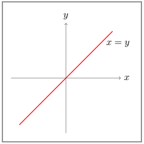
- Sets and Types
- Set Operations
- Set Comprehension

Sets

- 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. $x = y$)

Typing

- 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:

Important aspects of Typing

- 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)

Kinds of Types

- 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

The Z notation has only one built in type, integer:

$$\mathbb{Z} = \ldots -4, -3, -2, -1, 0, 1, 2, 3, 4, \ldots$$

It is an infinite set.

Natural numbers

- non-negative integers
- not a type in Z but a subset of its underlying type

$$\mathbb{N} = 0, 1, 2, 3, 4, \ldots$$

In some contexts it is useful to exclude zero from this set.

$$\mathbb{N}_1 = 1, 2, 3, 4, \ldots$$

- **Operations on integers:**
  The following operators are defined for the type integer and its subsets:

$+$	addition
-	subtraction
$*$	multiplication
div	integer division
mod	modulus

- **Note**
  The Z notation does not include
  - the set of real numbers
  - the set of characters

  as built-in types.

Basic Types

- Basic types are also called given sets.
- We could have two sets, written

$$[BOOK, PERSON]$$

- 'application-oriented' types
- The type **PERSON** frees us from any worry about what information we need about each person.
- elements of the type are uniquely identifiable.
- $[PERSON]$ is the set of all, uniquely identifiable persons.

- **All the PERSONs that ever lived, live now or will live in the future**

Free Types

- It can be very useful to define a type in terms of the possible values it can have.
- We can do this with a free type and the general format is:

$$FreeType ::= element1 \mid element2 \mid element3 \mid .... \mid elementn$$

- Examples

$$RESPONSE ::= yes \mid no$$
$$REPORT ::= InsertSucess \mid ElementAlreadyThere$$

- RESPONSE is a set containing the two values
  - yes
  - no
- We will cover free types more fully later.

All names designating values must be declared, i.e. their type must be stated. For example, to introduce a named value smallestCountry to be of the basic type *COUNTRY*, must write

$$[COUNTRY]$$
$$smallestCountry : COUNTRY$$

For the examples following, we will use countries by their abbreviation

Belguim	BE	Greece	EL	Lituania	LT	Portugal	P
Bulgaria	BG	Spain	ES	Luxembourg	LU	Romania	R
Czech Republic	CZ	France	FR	Hungary	HU	Slovenia	S

etc.

Truth Valued Statements

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

- A **Truth Valued Statement** is a statement whose value is either True or False.

- The following Truth Valued Statements is valid

$$smallestCountry = MT$$

Sets of Values

When a name is to be given to a set of values, the name is declared as a powerset of the type of the elements:

$$smallCountries : \mathbb{P}\, COUNTRY$$

This can be read "smallCountries is a subset of the set of COUNTRY's "

Set constants

A set of values is written by enumerating the set's values within braces

$$smallCountries = \{MT, LU, CY, SI, NL, EE, SK\}$$

Order is irrelevant i.e.

$$\{MT, LU, CY\} = \{LU, MT, CY\} = \{CY, MT, LU\}$$

Repeating a term does not matter;

$$\{MT, LU, CY\} = \{MT, MT, MT, LU, CY, MT\}$$

Enumerating Sets

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

The Empty Set
It is possible to have a set with no values. This is called the empty set. It is written

$\varnothing$ or
$\{\ \}$

A Singleton Set A set which contains one element is called a singleton set.

$$\{DE\}$$

Note that DE does not have the same type as $\{\mathrm{DE}\}$

Equivalence

Two values of the same type can be tested to check if they have the same value, by using the equals sign as in

$$x = y$$

Two sets are equal if they contain exactly the same elements. Note - order does not matter.

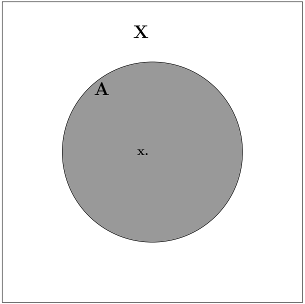
Non-equivalence

Similarly, two values of the same type can be tested to see if they are not the same by using the not equals sign, as in

$$x \neq y$$

Two sets are not equal if they do not contain exactly the same elements.

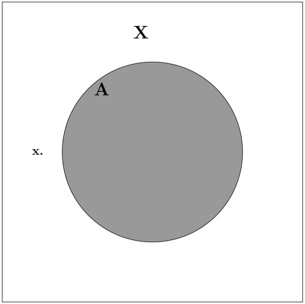
Set Membership



$[X]$
$A : \mathbb{P}\, X$
$x : X$
$x \in A$

Figure 1: Set membership

Set non-Membership



$[X]$
$A : \mathbb{P}\, X$
$x : X$
$x \notin A$

Figure 2: Set non-membership

Examples

Note

$$DE \in \{IE, BE, EE\}$$

is false

$$DE \notin \{IE, BE, EE\}$$

is true
but

$$1 \notin \{IE, BE, EE\}$$

is illegal because 1 in not of type COUNTRY

Size, Cardinality

The number of values in a set is called its size or cardinality and is signified by the hash sign, $\#$, acting as a function.

$\#\{DE, IE, EE\} = 3$
$\#\{DE\} = 1$
$\# IE$is illegal as IE is not a set
$\# \varnothing = 0$

Note: A set must be finite to legally apply the hash function to it.

Powersets

The powerset of a set S, is written

$$\mathbb{P}\, S$$

and is the set of all its subsets.

$$
\begin{array}{ll}
\mathbb{P}\{DE, FR, IE\} = & \\
\{\ \ \varnothing, & \text{the empty set} \\
\quad \{DE\}, \{FR\}, \{IE\}, & \text{all the singletons} \\
\quad \{DE, FR\}, \{FR, IE\}, \{DE, IE\}, & \text{all the pairs} \\
\quad \{DE, FR, IE\} & \text{all three elements} \\
\} &
\end{array}
$$

This means that a set of values of this powerset type is a subset of the underlying set type.

Finite sets

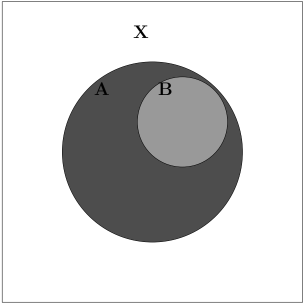
The cardinality of a set is only defined when the set is finite. This means that if we need to insist that a powerset is finite, we use the Finite set symbol

$$\mathbb{F}$$

e.g.

$$\mathit{someSetOfNumbers} : \mathbb{F}\mathbb{Z}$$

Set Inclusion



$[X]$
$A : \mathbb{P}\, X$
$B : \mathbb{P}\, X$
$B \subseteq A$

So,

$$\{1,2,3\} \subseteq \{1,2,3,4,5\}$$
$$\{1,2,3\} \subseteq \{1,2,3\}$$
$$\varnothing \subseteq \{1,2,3\}$$

In particular, note that the empty set is a subset of all sets.

Strict Inclusion

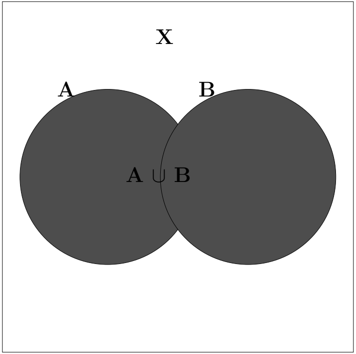
The operator

$$\subset$$

denotes strict inclusion or **is a proper subset of** i.e. the first set may not be equal to the second set.

$\{1,2,3\} \subset \{1,2,3,4,5\}$this is true
$\{1,2,3\} \subset \{1,2,3\}$this is false
$\{1,2,3\} \subseteq \{1,2,3\}$this is true
$\varnothing \subset \{1,2,3\}$this is true

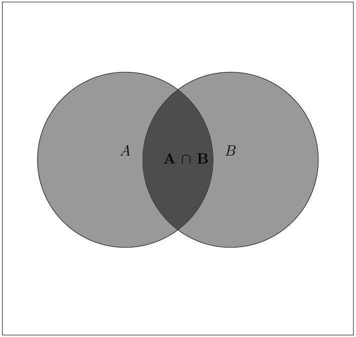
Set Union



$[X]$
$A : \mathbb{P}\, X$
$B : \mathbb{P}\, X$
$A \cup B$

Set intersection

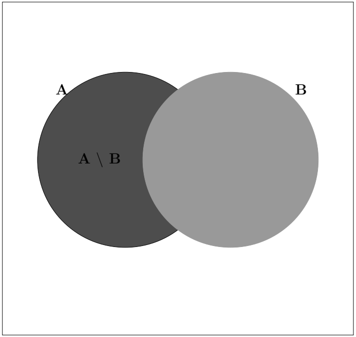
The union of two sets is the set containing all the elements that are in either the first set or second set or both.



$[X]$
$A : \mathbb{P}\, X$
$B : \mathbb{P}\, X$
$A \cap B$

Set Difference

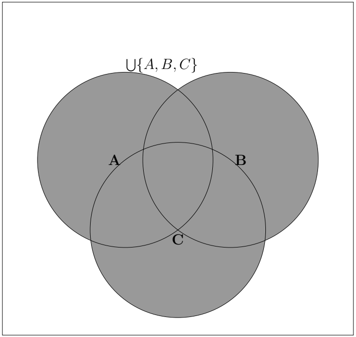
The set difference of two sets is the set containing all the elements of the first set that are not in the second.



$[X]$
$A : \mathbb{P}\, X$
$B : \mathbb{P}\, X$
$A \setminus B$

Distributed Union

Sometimes it is useful to be able to refer to the union of several sets or more precisely to a set of sets. This can be done with the distributed union operator.

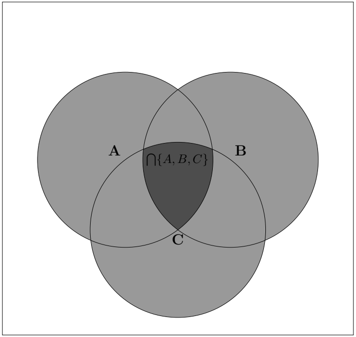


$[X]$
$A : \mathbb{P}\, X$
$B : \mathbb{P}\, X$
$C : \mathbb{P}\, X$
$\bigcup\{A, B, C\}$

Distributed Intersection

The distributed intersection of a set is the set of elements which are in all of the component sets.

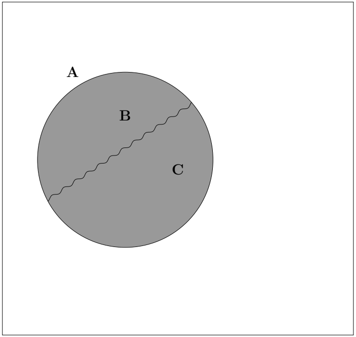
$$\bigcap\{\{DE, IE\}, \{IE, EE\}, \{BE, EE, GB, IE\}\} = \{IE\}$$



$[X]$
$A : \mathbb{P}\,X$
$B : \mathbb{P}\,X$
$C : \mathbb{P}\,X$
$\cap\{A, B, C\}$

Set partition

The union of two sets is the set containing all the elements that are in either the first set or second set or both.



$[X]$
$A : \mathbb{P}\,X$
$B : \mathbb{P}\,X$
$C : \mathbb{P}\,X$
$< B, C > \underline{partition}\ A$

Set partition

Note that

$$A = B \cup C$$
$$B \cap C = \varnothing$$

In this case, B and C are said to be disjoint i.e. they have no elements in common.

$$\underline{disjoint} < B, C >$$

Also, the distributed union of B and C makes up another set, A. The sets B and C are said to partition the set A because

$$A = B \cup C \; and$$
$$B \cap C = \varnothing$$

Set partition

More than two sets can partition a set. This can be a useful way of describing interset relationships. More formally, this is known as partition

$$< B, C > \underline{partition}\ A$$

Ranges of numbers

- The range of values m..n is the set of all integers between and including m and n, e.g.

$$2..5 = \{2, 3, 4, 5\}$$

- The notation m..n therefore denotes a valid set and can be used as such.

Set construction

- Up to now, we have needed to enumerate the elements in a set.
- Another way of constructing a set is using set comprehension i.e. don't explicitly enumerate the elements, instead define the set as the set of all elements that obey a given set of rules.
- Set comprehension is a more powerful and elegant way of constructing sets.

Set Comprehension Notation

The set of values

$$\{1, 2, 3, 4, 5, 6, 7, 8\}$$

could also be written using set comprehension

$$\{x : \mathbb{N} \mid 1 \leq x \leq 8 \bullet x\}$$

The structure is below:

$$\begin{aligned} Set - Exp = \{&declaration \mid constraint(or\ TruthValuedStatement) \\ &\bullet expression\ or\ term\} \end{aligned}$$

Example

$$Squares = \{x : \mathbb{N} \mid 1 \leq x \leq 20 \bullet x^2\}$$

is the set of squares of every integer from 1 to 20.

Set Comprehension Notation

The members of the set

$$\{Declaration \mid constraint \bullet expression\}$$

are the values taken by the *expression* when the *variables* introduced by the *declaration* take all possible values which make the *constraint* true.
The above set Squares might be read as:

Select all those integers which lie between 1 and 20 inclusive, and form the set of their squares ".

In future examples, we will use more advanced constraints. Any valid Truth Valued statement is valid here.

Omissions

If the constraint is omitted it is taken to be True i.e. it does not constrain the variables in the expression.

$$Squares = \{x : \mathbb{Z} \bullet x^2\}$$

is the set of squares of all integers.

Tuples and Product Types

- A tuple is an ordered collection of two or more objects.
- Examples are common in records :
  - (Name, Address) is an ordered pair
  - in such mathematical constructs as vectors and co-ordinate points
  - If we add a third component, PhoneNo we get an ordered triple (Name, Address, PhoneNo), and so on.

Characteristic Tuple

The characteristic tuple is the tuple formed from the variables in the declaration part of a set comprehension, in the order of declaration. In

$$K = \{a : \mathbb{N}, b : \mathbb{N} \mid a \leq 10 \wedge b \leq 10\}$$

the characteristic tuple is (a,b).
The default Expression in a set comprehension is the characteristic tuple of the set:

$$K = \{a : \mathbb{N}, b : \mathbb{N} \mid a \leq 10 \wedge b \leq 10 \bullet (a, b)\}$$

You should use the latter version.

Any questions?