

Reviewing Set Theory

COMP2600 — Formal Methods for Software Engineering

Jeremy Dawson

Australian National University

Semester 2, 2011

Why should we study set theory?

Understanding mathematics: It is the foundation!

Programming: Types and classes are sets.

Specifying software: Legal inputs and appropriate outputs are sets.

Reasoning about programs: This inevitably relies on sets of data values.

Semantics: The meaning of a program is a mapping of one set of data to another.

Set Theory Basics

Basic concepts:

- Objects, elements, sets
- Enumeration notation: as in $\{1, 2, 3\}$
- Is-element-of notation: as in $5 \in \{1, 3, 5, 7\}$

Set equality and notational ambiguity:

- $\{1, 2, 3\} = \{3, 2, 1\}$
- $\{1, 2, 3, 2, 1\} = \{1, 2, 3\}$

Standard sets

Named sets

- **numbers:**
 \mathbb{N} is the set of natural numbers; \mathbb{Z} is the set of integers;
 \mathbb{Q} is the set of rational numbers; \mathbb{R} is the set of real numbers.
- **booleans:** The set $\{\mathbf{T}, \mathbf{F}\}$ is typically called **Bool**.
- **characters:** Often consisting of the set of ASCII characters, and often will be referred to as **Char**.

The empty set \emptyset

- Basic properties: $\forall x. x \notin \emptyset$ and $\forall A. \emptyset \subseteq A$
- Note: $\{\emptyset\}$ is not the same set as \emptyset .

Notation for Infinite Sets

The “*dot-dot-dot*” notation

- $\{1, 3, 5, 7, \dots\}$ ‘obviously’ denotes the set of odd numbers.

Problems:

- Is it adequate?
Occasionally, but in general, no.
- Is it ambiguous?
Is $\{1, 3, \dots\}$ the same set as $\{1, 3, 4, \dots\}$?

Characteristic Predicates

Need a better notation:

- $S = \{x \in D \mid P(x)\}$
- P will be called the characteristic predicate of the set S .

We often write $\{x \mid P(x)\}$ when the domain is understood.

Given a set S , we can get the characteristic predicate P from S :
we *define* $P(x)$ to be true iff $x \in S$

We should ask ourselves “Is this always a sensible notation?”

Russell's Paradox

Reveals a problem with an unconstrained notion of sets: even if P is a sensible predicate, it doesn't necessarily characterise a set.

Russell considered the set of all sets that are not members of themselves:

$$S \equiv \{A \mid A \notin A\}$$

and posed the question "Is S a member of itself?"

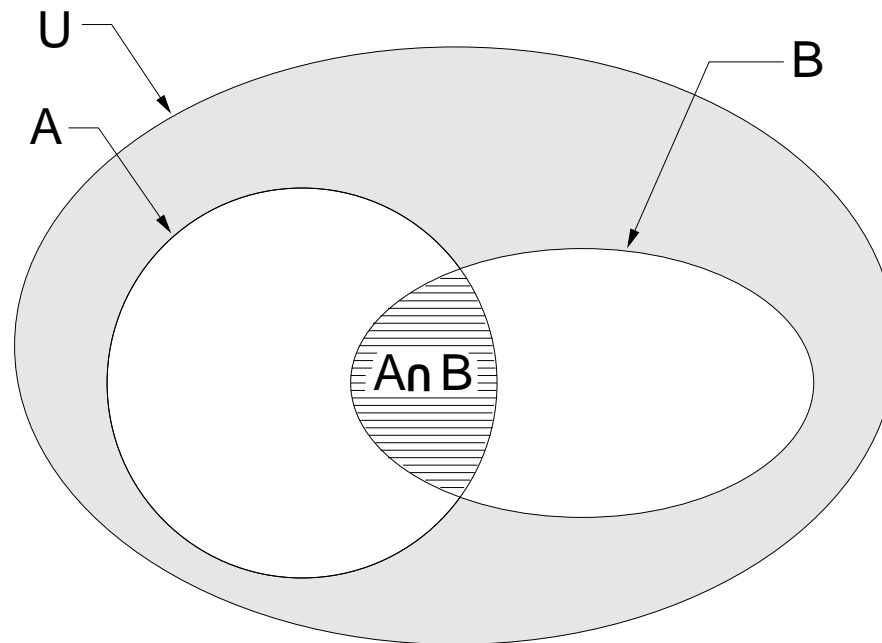
Consider the 2 cases:

- Suppose $S \in S$. Then S is in the set $\{A \mid A \notin A\}$, so $S \notin S$.
- Suppose $S \notin S$. Then S does not satisfy the predicate $A \notin A$, so $S \in S$.

Hence there is no such S . The predicate is sensible, but it does not characterise a set.

Universal Sets

Venn diagrams are a notation where union and intersection can be well illustrated. If complements are to have any meaning then Venn diagrams often show a box that corresponds to the whole domain of discourse — the universal set.



New sets from Old

Subset:

- Notation: $A \subseteq B$ holds iff $\forall x \in A. x \in B$
- If P is a predicate of appropriate type, and A is given, then $\{x \in A \mid P(x)\}$ is a new set.
Of course it is a subset of A .

On subsequent slides we recall other ways to build sets.

- Power set
- Cartesian Product
- Union and intersection

New sets from Old - II

Power set:

- Notation: $\mathcal{P}(A)$ denotes the power set of A .
- Definition: $\mathcal{P}(A) \equiv \{s \mid s \subseteq A\}$
i.e. $\mathcal{P}(A)$ is the set of all subsets of A .
- Example: $\mathcal{P}(\{1,2\}) = \{\emptyset, \{1\}, \{2\}, \{1,2\}\}$

Cardinality notation: $|A|$ indicates the cardinality of A .

The power set is so called because $|\mathcal{P}(A)| = 2^{|A|}$

Cartesian Product

Notation:

- $A \times B = \{(x, y) \mid x \in A \wedge y \in B\}$
- $A \times B \times C = \{(x, y, z) \mid x \in A \wedge y \in B \wedge z \in C\}$
- etc.

Apparent abuse of notation: The product is defined in terms of a set of ordered pairs. Since the set they belong to is the Cartesian product, the definition seems circular.

Resolution: The existence of the Cartesian product is an axiom of set theory. Consequently the expression $\{(x, y) \in A \times B \mid x \in A \wedge y \in B\}$ is well formed.

The Usual Binary Operations

Binary union and intersection (of 2 sets belonging to a universe of discourse):

- $A \cup B = \{x \in U \mid (x \in A) \vee (x \in B)\}$
- $A \cap B = \{x \in U \mid (x \in A) \wedge (x \in B)\}$

U is the universe of discourse. That is, $A \subseteq U$ and $B \subseteq U$.

Union and Intersection of a set of sets

- Union: $\bigcup_{i \in I} D_i = \{x \in U \mid \exists i \in I. x \in D_i\}$
- Intersection: $\bigcap_{i \in I} D_i = \{x \in U \mid \forall i \in I. x \in D_i\}$

I is called the index set.

Example:

$$I = \{0, 1, \dots, 26\}$$

Let D_i be the set of the first i letters of the alphabet, eg, $D_4 = \{a, b, c, d\}$

Then $\bigcup_{i \in I} D_i = \{a, b, \dots, z\}$ and $\bigcap_{i \in I} D_i = \emptyset$

And if $J = \{2, 4, 7\}$, then

$$\bigcup_{i \in J} D_i = \{a, b, c, d, e, f, g\} \quad \text{and} \quad \bigcap_{i \in J} D_i = \{a, b\}$$

Relations as Sets

Set-theoretic view: relations and functions are both sets of ordered pairs.

Relations:

- Any subset of $A \times B$ is a relation between elements of A and of B .
- That is, any $R \in \mathcal{P}(A \times B)$ is a relation between A and B .
- We often write xRy to mean $(x, y) \in R$
- The characteristic predicate P of R is given by:
 $P(x, y)$ is true iff $(x, y) \in R$
- There is a curried version of P , call it Q : $Q x y$ is true iff $(x, y) \in R$
- in Haskell, the infix notation (eg, $x < y$) works where $<$ is the curried version of the characteristic predicate

Flavours of Relations

- R is *reflexive* if, for all $x \in A$, $(x, x) \in R$. That is, xRx .
- R is *transitive* if $(xRy \wedge yRz)$ implies xRz .
- R is *anti-symmetric* if, whenever xRy and yRx then $x = y$.

Functions as Sets

Functions:

- Each function f from A to B , is also an element of $\mathcal{P}(A \times B)$, but subject to the condition that if $(x, y_1) \in f \wedge (x, y_2) \in f$ then $y_1 = y_2$.

Reference Details

Chapter 5 of Grassman and Tremblay is called *Sets and Relations*.

5.1 **Sets and Set Operations**. You must know this.

5.2 **Tuples, Sequences and Power-sets**.

5.2.1 **Introduction**: should know.

5.2.2 **Tuples and Cartesian Products**: must know.

5.2.3 **Sequences and Strings**: We'll get to this.

5.2.4 **Power-sets**: should know.

5.2.5 **Types and Signatures**: We'll cover this.

5.3 **Relations**: We'll be talking about this.

5.4 **Properties of Relations**: We'll be talking about this.