# Lecture 3
# Sets, Relations, Functions

**David Garlan**

**Carnegie Mellon University**

**Models of Software Systems**

**15-671**

**Fall 1995**

# The Story So Far

- **Formal Systems**
  - > **Syntax**
    - » language: alphabet + grammar
    - » deductive system: axioms + rules of inference
  - > **Semantics**
    - » interpretation
- **Propositional logic**
- **Predicate logic (with equality)**
- **Natural deduction**
- **New theories**

# This Lecture

- In this lecture we will develop one of the most important theories for modelling software systems
  > set theory
  >
  > relations
  >
  > functions

# Sets

- Collection of distinct objects
- In this course elements of well-formed sets are drawn from a larger homogeneous set.
- Examples:
  > {1, 3, 5, 7, ... }
  > {red, green, blue, green}
  > {Peter, Doug, Kevin, Eduardo, ... }
  > { yes, no }
- But not: {red, 1, 2, 3 }

?=? {blue, red, green}

# Set Definition & Membership

- **Sets can be defined by enumeration**
  - Odds == {1, 3, 5, 7, ... }
  - Colors == {red, green, blue, blue, green}
  - 15-671 == {Elizabeth, Doug, Kevin, Eduardo, ... }
  - > Note the use of "==" for definition ($\widehat{=}$)
- **Membership test: 3 $\in$ Odds ; 2 $\notin$ Odds**
  - > note $\in$ is a predicate over sets and elements
- **Some sets have predefined names**
  - $\emptyset$ the set with no elements (the "null" or "empty" set) -- also written {}
  - $\mathbb{N}$ the set of natural numbers {0, 1, ... }
  - $\mathbb{Z}$ the set of integers {... -2, -1, 0, 1, 2, ... }

# Given Sets

- We can define new sets of primitive elements
  - > Called “basic sets”
  - > Written [TheSet]
  - > We know nothing about the elements, but assume there is an = operator over them
- Examples
  - > [BookIdentifiers]
  - > [Date,Name,Place]
- Later we will see how we can additional assertions about the elements of a new given set.

# Set Equality and Cardinality

- **Sets have an equality operator.**

  **Two sets are equal if they have the same elements. That is:**

  $(S = T) \Leftrightarrow (\forall x \bullet x \in S \Leftrightarrow x \in T)$

- **Note this implies that ordering of elements does not matter.**

  **Qn: Why?**

- **Cardinality (size) of a set: #**

  $> \# \{1, 2, 4\} = 3$

  $> \# \{ \{1,2\}, \{1,2,3,4,5,6,7\} \} = ?$

  $> \# \{1, 2, 2, 4\} = ?$

# Set Operators

- Can form new sets from other sets using operators:
  - > $\cap$ (intersection), $\cup$ (union), \ (difference)
- Examples: Let A == {1,2,3} and B == {3,4,5}
  - > $A \cap B = \{3\}$
  - > $A \cup B = \{1,2,3,4,5\}$
  - > $A \setminus B = \{1,2\}$
- Note the use of "=" versus "==".

**Qn: Why?**

- A is a subset of B ( $A \subseteq B$) if every element of A is also an element of B.
- $A = B$ if $A \subseteq B$ and $B \subseteq A$.

**Qn: Why?**

# Set Axioms & Laws

- **Basic axioms**
  - > Set membership: $\forall x \bullet x \in \{x\}$
  - > Empty set: $\forall x \bullet \sim(x \in \varnothing)$
- **Laws (can be proven)**
  - > $A \cap B = B \cap A$ $\quad\quad$ $A \cup B = B \cup A$
  - > $(A \cup B) \cup C = A \cup (B \cup C)$
  - > $(A \cap B) \cap C = A \cap (B \cap C)$
  - > $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
  - > and lots of others

**Qn: How would you prove these laws?**

# Set Comprehension

- Enumerating all of the elements of a set is not always possible.
- Would like to describe a set by in terms of a distinguishing property of its elements.
  - > Roster == the set of students in 15-671
  - > Pgh == the set of residents of Pittsburgh
  - > Primes == the set of integers that are prime
- Each element "satisfies"some "criterion".

  Qn: How can we define such criteria?

  Ans: Predicates!
- This kind of set specification is called *set comprehension*

# Set Comprehension (2)

- Simple form of set comprehension

  $\{x : S \mid P(x)\}$

  "the set of all x in S that satisfy P(x)", or

  "the set of all x in S such that P(x)"

- Examples
  - > natural numbers less than 20: $\{x: \mathbb{N} \mid x < 20\}$
  - > even integers: $\{x: \mathbb{Z} \mid (\exists\, y: \mathbb{Z} \bullet x = 2y)\}$
    - » note we identify the types of variables in predicates
  - > all natural numbers: $\{x: \mathbb{N} \mid true\}$
  - > empty set of natural numbers: $\{x: \mathbb{N} \mid false\}$

# Set Comprehension (3)

- Sometimes it is helpful to specify a "pattern" for the elements.
  - > We will use the form: {x: S • f(x)}, where f is some function defined on elements of S.
  - > Note the use of "• " instead of "|"
- Examples:
  - > squares: $\{x: \mathbb{N} \bullet x^2\}$
  - > authors of books in library:

    {b: Library • author-of(b) }

    *where* author-of: Book $\rightarrow$ Author
  - > birthdays of people in 15-671

    {person: 15-671• birthday-of(person) }

Qn: How might this have been written using just predicates?

# Alternative Notation

- In more traditional mathematics the expression part is usually written first
- Examples
  - $\{x : \mathbb{N} \bullet x^2\}$ would be written $\{ x^2 \mid x \in \mathbb{N} \}$
- For reasons that will be come clear later in the course we do not do this.

# Set Comprehension (4)

- **Most general form combines the two forms**

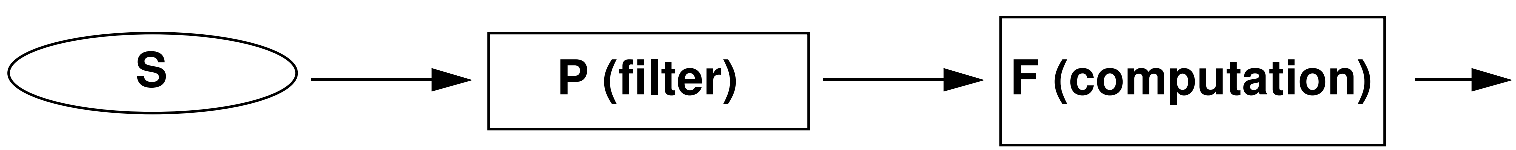
  $\{x: S \mid P(s) \bullet f(x)\}$

- **Examples**
  - > squares of integers less than 20:

    $\{x: \mathbb{N} \mid x < 20 \bullet x^2\}$
  - > squares of even integers:

    $\{x: \mathbb{Z} \mid (\exists\, y: \mathbb{Z} \bullet x = 2y) \bullet x^2\}$

- **Can think of this as "nested pipelines"**



# Power Sets

- The set of all subsets of S is referred to as the *power set* of S and written $\mathbb{P}$ S.
- Examples:
  - $\mathbb{P}$ {1, 2, 3 } =
    { ∅, {1}, {2}, {3}, {1,2}, {1,3}, {2,3}, {1,2,3}}
  - $\mathbb{P}$ $\mathbb{N}$ = { ∅, {1},{1,2}, {1,3} ...}
- Power sets can be used to define new types, and can be used in declarations
  - class-groups == $\mathbb{P}$ Student
  - Integer-sets == $\mathbb{P}$ $\mathbb{N}$
  - If x : $\mathbb{P}$ $\mathbb{N}$, then x is a *set of* Integers.
  - Whenever you see " .. : P .." read "set of"
  - Alternatively could write: x: Integer-sets

Note difference between ":" and "∈"

# Cartesian Products

- **Ordered pairs, triples, etc.**
- **Examples:**
  - > **(2, 3) $\neq$ (3, 2)**
  - > **S == { (2, red), (5, blue), (3, red) }**
- **The set of all tuples constructed from two sets is called a *Cartesian Product*, or just *Product* of those sets.**
  - > **If S and T are those sets, this is written S x T**
- **Examples**
  - > $\mathbb{N}$ **x** $\mathbb{N}$ **pairs of natural numbers**
  - > **(2, red):** $\mathbb{N}$ **x Color**
  - > **S:** $\mathbb{P}$ **(**$\mathbb{N}$ **x Color)**

# More Tuples

- In general tuples can be described using variables, much like the record constructor of Pascal
- General form:
  - > {declarations | predicate}
  - > the predicate is sometimes called an invariant over the state space defined by the declarations
- Examples
  - > $\{(x, y): \mathbb{N} \times \mathbb{N} \mid y = x + 1\} = \{(0,1), (1,2), (2,3), \ldots\}$
  - > $\{x, y: \mathbb{N} \mid y = x + 1\} = \{(0,1), (1,2), (2,3), \ldots\}$
  - > $\{x: \mathbb{P}\, \mathbb{Z}; y: \mathbb{N} \mid y = \#x\} = \{ (\{-1,2,3\}, 3), ( \varnothing, 0)\ldots\}$

# Relations

- A *relation* is a set of pairs.
- Examples:
  > A == { (1,1), (1,2), (2,2) }
  >
  > B == { (2, red), (5, blue), (3, red) }
  >
  > C == { (David, Jun 1}, (Mary, Aug 2), (Bill, Feb 5) }
- The set of all relations over sets S, T is indicated by <u>S $\leftrightarrow$ T</u>
  > (Note S $\leftrightarrow$ T is equivalent to $\mathbb{P}$ (S x T).)
- Examples:

  A: $\mathbb{N} \leftrightarrow \mathbb{N}$ B: $\mathbb{N} \leftrightarrow$ Color C: Person $\leftrightarrow$ Date

# Relations (2)

- The *domain* of a relation is the set of first elements. ("dom")
- The *range* of a relation is the set of second elements. ("ran")
- Examples:

A == { (1,1), (1,2), (2,2) }

dom A = { 1, 2 } and ran A = { 1, 2 }

even though A $\in \mathbb{N} \leftrightarrow \mathbb{N}$

B == { (2, red), (5, blue), (3, red) }

dom B = { 2,3,5 } and ran B = { red, blue }

C == { (David, Jun 1}, {Mary, Aug 2}, {Bill, Feb 5) }

dom C = { David, Mary, Bill}

ran C = { Jun 1, Aug 2, Feb 5 }

# Functions

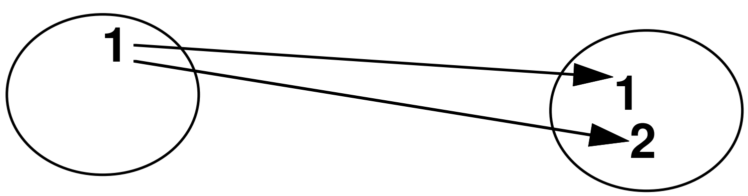
- A function is a relation such that no two distinct pairs contain the same first element.
- Examples:

  B == { (2, red), (5, blue), (3, red) }

  C == { (David, Jun 1}, {Mary, Aug 2}, {Bill, Feb 5) }

  <u>No</u>:

  A == { (1,1), (1,2), (2,2) }

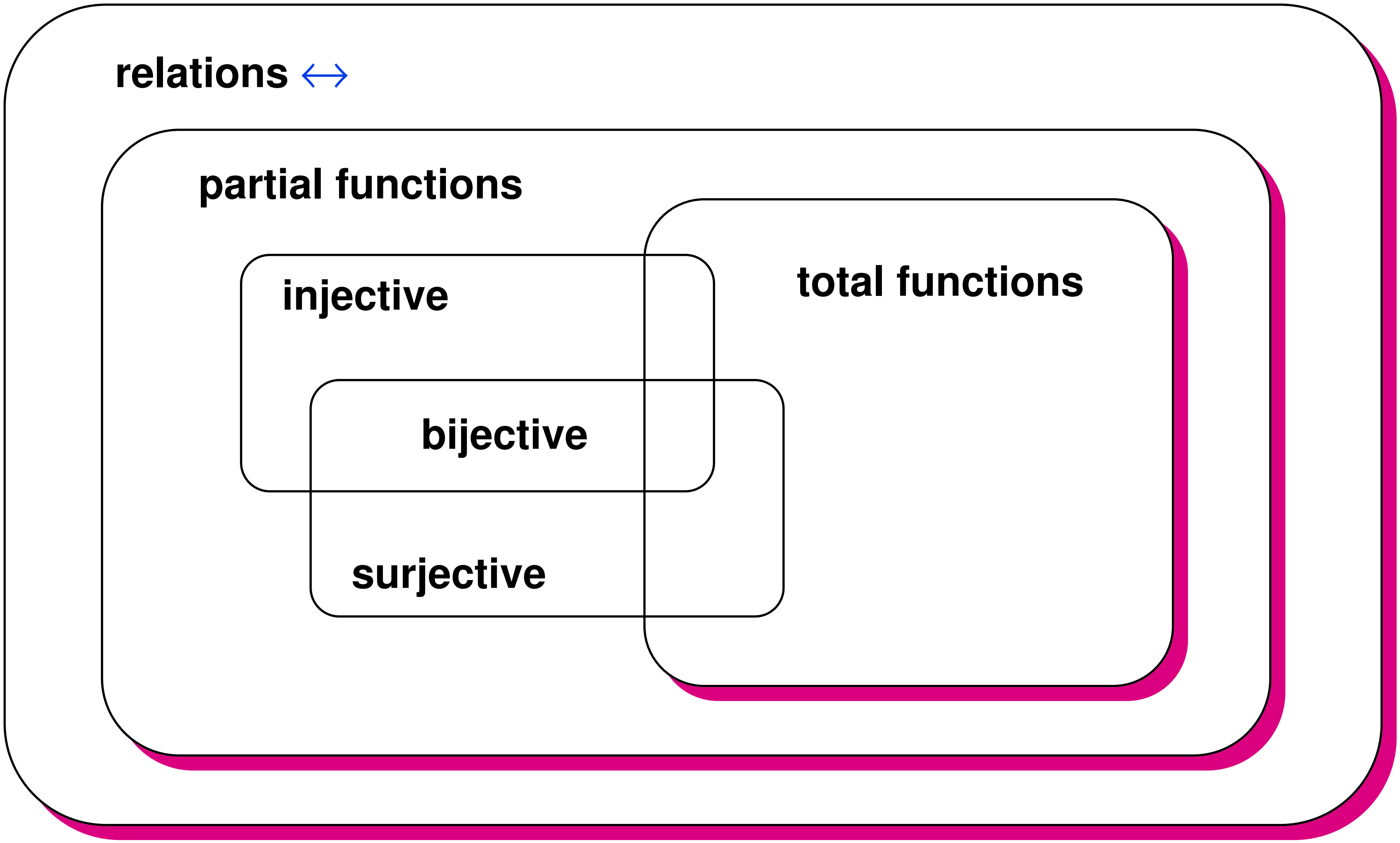


# Some special cases

**Suppose f: $A \leftrightarrow B$**

1. **f is a function defined for *all* values of A**
   we say f is a "total" function, and write $A \rightarrow B$
2. **f is a function defined for *some* values of A**
   we say f is a "partial" function, and write $A \rightarrow B$
3. **f is a function defined for a *finite set* of values of A**
   we say f is a "finite" function, and write $A \rightarrow B$
4. **f is a function for which no element in ran(f) is associated with more than one element in dom(f)**
   we say f is a "one-to-one" or "injective" function, and write $A \rightarrow B$
5. **f is a function whose range is B**
   we say f is an "onto" or "surjective" function, and write $A \rightarrow B$
6. **f is both one-to-one and onto**
   we say f is a "bijection" function, and write $A \rightarrow B$

# Special Cases (2)



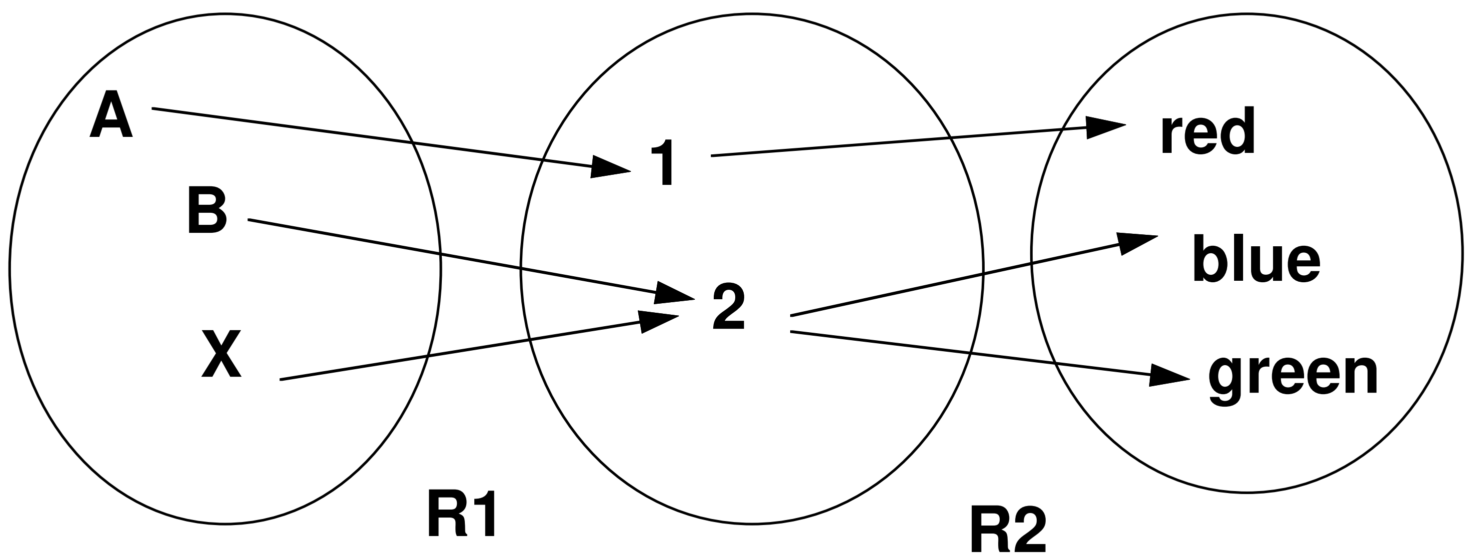
# Relations/Functions as Sets

- Since relations are just sets (of pairs) we can apply set operators to them.
- Example:
  - R1 = {(1,red), (2,blue)}
  - R2 = {(3,green), (2,blue)}
  - R1 $\cup$ R2 = {(1,red), (2,blue), (3,green)}
  - # R1 = 2

Question: is the union/intersection of two functions a function?

# Relational Composition

- If the range of one relation is the domain of another can form the composition (R1;R2)



# Overwriting

- Frequently we will want to change the value of a function for one or more values
- The *overwriting* operator does this:
- Example:
  - $f = \{(1,red), (2,blue), (3,green)\}$
  - $g = \{(1,pink), (4,mauve)\}$
  - $f \oplus g = \{(1,pink), (2,blue), (3,green), (4,mauve)\}$

**Note: replacement only considers domain values**

# Other Operators

- There is a rich library of other auxiliary definitions.
- We'll be encountering many as we proceed through the class
- Some examples:
  - > $\mathbb{N}$, the set of natural numbers
  - > $\mathbb{N}_1$, the set of positive natural numbers
  - > .., the "between" operator: 2..5= {2,3,4,5}

# Last Slide

# More Functions

- *Partial* versus *total* functions
- Example:
  - > sqroot is a partial function from $\mathbb{N}$ to $\mathbb{N}$
  - > square is a total function from $\mathbb{N}$ to $\mathbb{N}$
- *One-to-one* (injective) functions.
- *Onto* (surjective) functions
- *Bijection* (1-1, onto)
- *Finite* functions

# Fancy Function Symbols

- A number of special symbols have been defined for relations and functions:
  - > relations $A \leftrightarrow B$
  - > total functions $A \rightarrow B$
  - > partial functions $A \rightarrow B$
  - > finite functions $A \rightarrow B$
  - > onto functions $A \rightarrow B$
  - > 1-1 functions $A \rightarrow B$
  - > bijection $A \rightarrow B$

  and combinations of these