دانشگاه آزاد اسلامی واحد تبریز

نام درس: روش های رسمی در مهندسی نرم افزار

بخش: مجموعه ها، رابطه ها و توابع

نام استاد: دکتر مسعود کارگر

مجموعه

- Collection of distinct objects
- Each set's objects are drawn from a larger *domain* of objects all of which have the same type --- sets are homogeneous
- Examples:

{2,4,5,6,...}	set of integers
{red, yellow, blue}	set of colors
{true, false}	set of boolean values
{red, true, 2}	for us, not a set!

domain

ارزش یک مجموعه

- Is the collection of its members

- Two sets A and B are equal iff
  – every member of A is a member of B
  – every member of B is a member of A

- x є S denotes “x is a member of S”
- Ø denotes the empty set

تعریف مجموعه

- We can define a set by enumeration
  - PrimaryColors == {red, yellow, blue}
  - Boolean == {true, false}
  - Evens == {…, -4, -2, 0, 2, 4, …}

- This works fine for finite sets, but
  - what do we mean by "…" ?
  - remember, we want to be precise

تعریف مجموعه

- We can define a set by *comprehension*, that is, by describing a property that its elements must share
- Notation: { x : D | P(x) }
  – Form a new set of elements drawn from domain D by including exactly the elements that satisfy predicate (i.e., Boolean function) P
- Examples:

{ x : N | x < 10} — *Naturals less than 10*

{ x : Z | (∃ y : Z | x = 2y) } — *Even integers*

{ x : N | x > x} — *Empty set of natural numbers*

کاردینالیتی یا تعداد اعضای مجموعه

- The *cardinality* (#) of a finite set is the number of its elements

- Examples:
  - # {red, yellow, blue} = 3
  - # {1, 23} = 2
  - # Z = ?

- Cardinalities are defined for infinite sets too, but we'll be most concerned with the cardinality of finite sets

عملگرهای مجموعه

- Union (X, Y sets over domain D):
  - $X \cup Y \equiv \{e: D \mid e \in X \text{ or } e \in Y\}$
  - {red} ∪ {blue} = {red, blue}

- Intersection
  - $X \cap Y \equiv \{e: D \mid e \in X \text{ and } e \in Y\}$
  - {red, blue} ∩ {blue, yellow} = {blue}

- Difference
  - $X \setminus Y \equiv \{e: D \mid e \in X \text{ and } e \notin Y\}$
  - {red, yellow, blue} \ {blue, yellow} = {red}

زیرمجموعه

- A *subset* holds elements drawn from another set
  - X ⊆ Y iff every element of X is in Y
  - {1, 7, 17, 24} ⊆ Z
- A *proper subset* is a non-equal subset
- Another view of set equality
  - A = B iff (A ⊆ B and B ⊆ A)

مجموعه‌های توان

- The power set of set S (denoted *Pow* (S)) is the set of all subsets of S, i.e.,
  - Pow (S) ≡ {e | e ⊆ S}

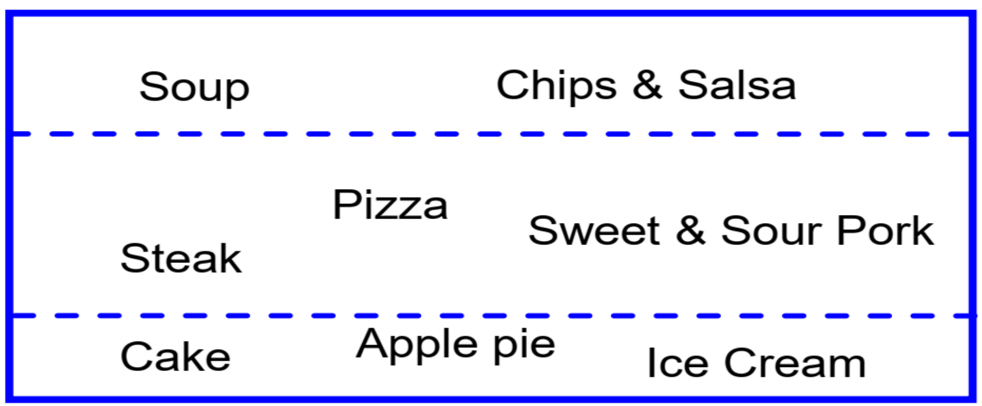
- Example:
  – *Pow* ({a,b,c}) = {Ø, {a}, {b}, {c}, {a,b}, {a,c}, {b,c}, {a,b,c}}

Note: for any S, Ø ⊆ S and thus Ø ∈ Pow(S)

بخش‌بندی مجموعه‌ها
Set Partitioning

- Sets are *disjoint* if they share no elements
- Often when modeling, we will take some set S and divide its members into disjoint subsets called *blocks* or *parts*

- We call this division a *partition*
- Each member of S **belongs to exactly** one block of the partition



مثال برای بخش‌بندی مجموعه‌ها

Model residential scenarios

- Basic domains: *Person*, *Residence*

- Partitions:
  – Partition *Person* into *Child*, *Adult*
  – Partition *Residence* into *Home, DormRoom, Apartment*

(Relationships) بیان روابط

- It's useful to be able to refer to structured values
  – a group of values that are bound together
  – e.g., struct, record, object fields

- Alloy is a calculus of *relations*
- All of our Alloy models will be built using relations (sets of tuples)
- … but first some basic definitions

ضرب (product)

- Given two sets A and B, the product of A and B, usually denoted A x B, is the set of all possible pairs (a, b) where a ∈ A and b ∈ B

$$A \times B \equiv \{ (a, b) \mid a \in A, b \in B \}$$

- Example: PrimaryColor x Boolean:

{

(red,true),	(red, false),
(blue,true),	(blue, false),
(yellow, true),	(yellow, false)

}

رابطه (Relation)

- A binary relation R between A and B is an element of
- *Pow*(A x B), i.e., $R \subseteq A \times B$

- Examples:

- – Parent : Person x Person
  - Parent = { (John, Autumn), (John, Sam) }

- – Square : Z x N
  - Square = {(1,1), (-1,1), (-2,4)}

- – ClassGrades : Person x {A, B, C, D, F}
  - ClassGrades = { (Todd,A), (Jane,B) }

(Relation) رابطه

- A ternary relation R between A, B and C is an element of *Pow*(A x B x C)

- Example:
  – FavoriteSoftDrink : Person x SoftDrink x Price
    • FavoriteSoftDrink = { (John, Fanta, \$4), (Ted, Soda, \$2), (Steve, Soda, \$2) }

- N-ary relations with n>3 are defined analogously (n is the arity of the relation)

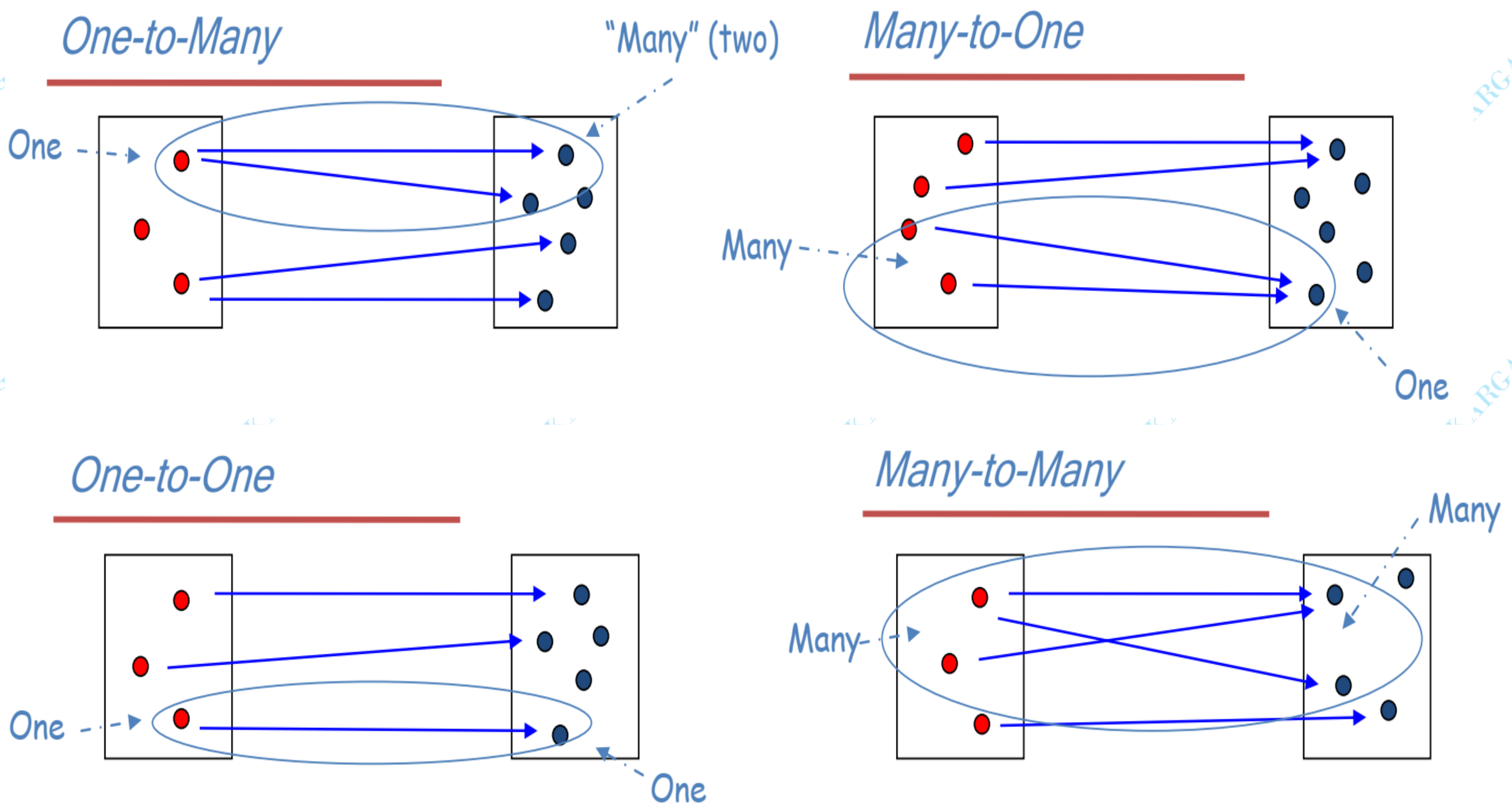
(Binary Relations) رابطه دوتایی

- The set of first elements is the *definition domain* of the relation
  – Parent = { (John, Autumn), (John, Sam) }
  – *domain*(Parent) = {John}   NOT Person!

- The set of second elements is the *image* of the relation
  – *image*(Square) = {1,4}   NOT N!

- How about {(1,blue), (2,blue), (1,red)}
  – domain?   image?

ساختارهای متداول رابطه‌ها



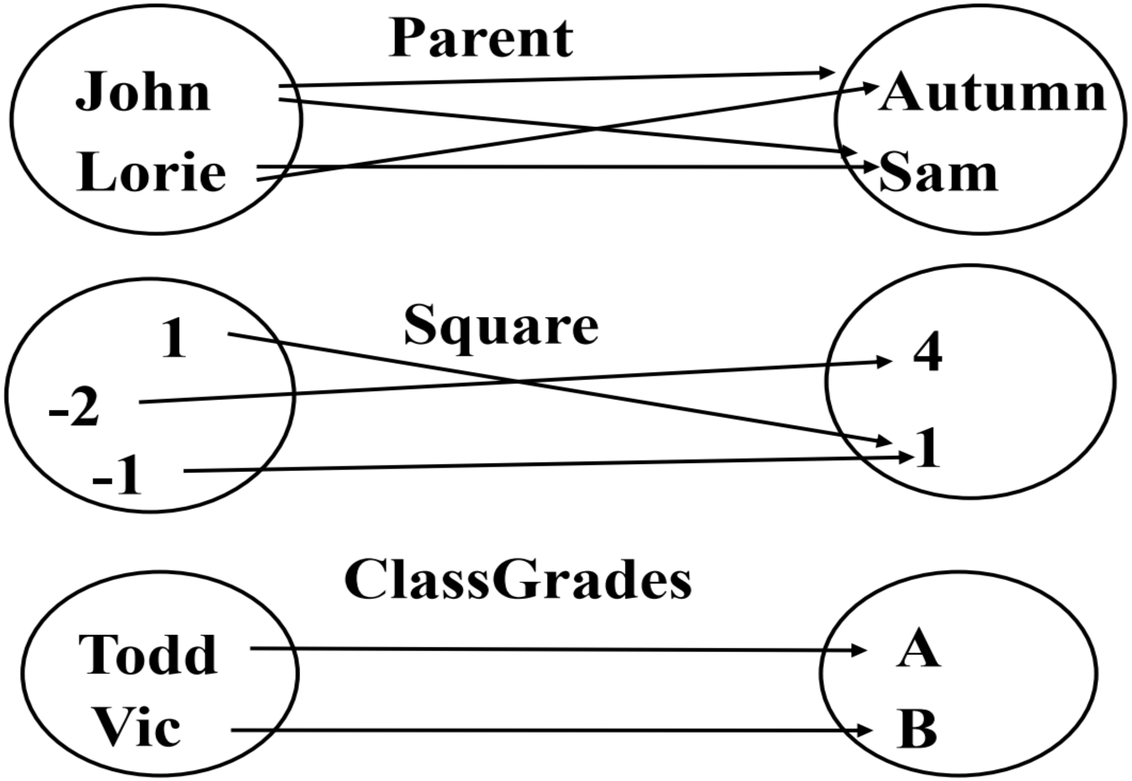
توابع

- A *function* is a relation F of arity n+1 containing no two distinct tuples with the same first n elements,
  - i.e., for n = 1,
    $\forall$ (a1, b1) $\in$ F, $\forall$ (a2, b2) $\in$ F, (a1 = a2 $\Rightarrow$ b1 = b2)

- Examples:
  - { (2, red), (3, blue), (5, red) }
  - { (4, 2), (6,3), (8, 4) }

- Instead of F: A1 x A2 x … x An x B
  we write F: A1 x A2 x … x An -> B

رابطه در مقابل توابع



Many-to-many

Many-to-one

One-to-one

In other words, a function is a relation that is X-to-one.

انواع توابع

- Consider a function f from S to T

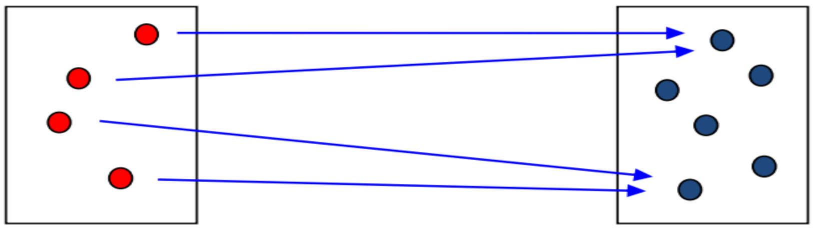
- f is ***total*** if defined for all values of S
- f is ***partial*** if undefined for some values of S

- Examples
  - Squares : Z -> N, Squares = {…, (-1,1), (0,0), (1, 1), (2,4), …}

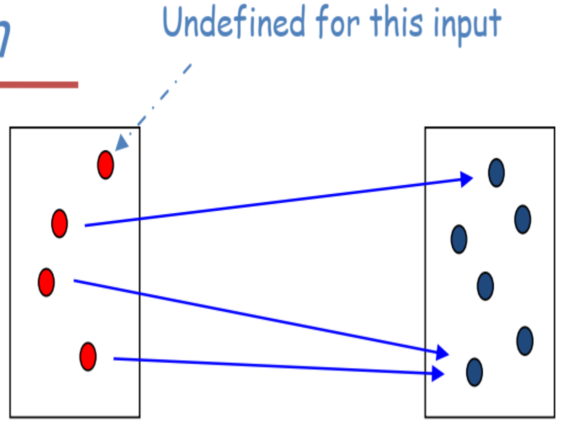
  - Abs = { (x, y) : Z x N | (x < 0 and y = -x) or (x ≥ 0 and y = x) }

ساختارهای توابع

Total Function



Partial Function

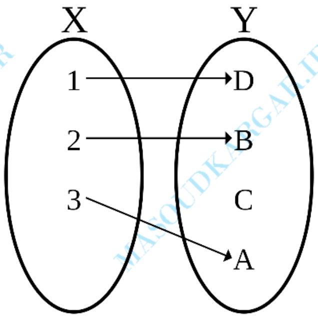


Note: the empty relation over an non-empty domain is a partial function

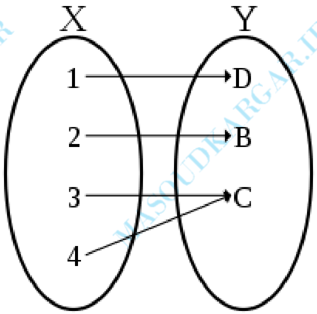
انواع توابع

A function f: S -> T is

- ***injective*** (*one-to-one*) if no image element is associated with multiple domain elements

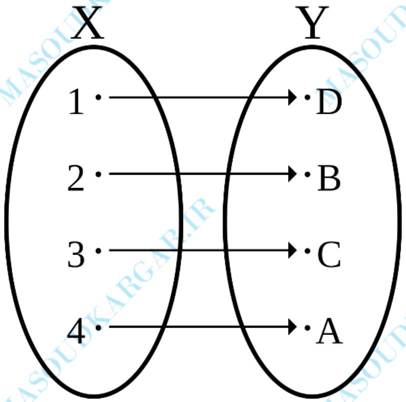


- ***surjective*** (*onto*) if its image is T



انواع توابع

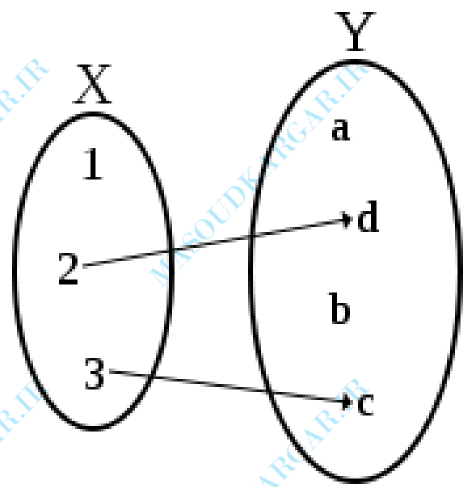
- *bijective* if it is both injective and surjective



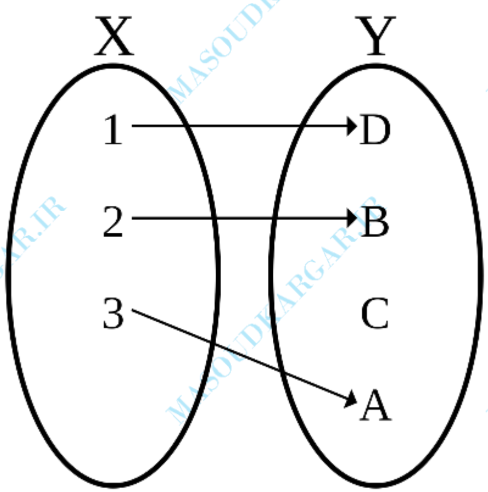
We’ll see that these come up frequently
– can be used to define properties concisely

تابع جزئی و تابع کلی

Partial Functions

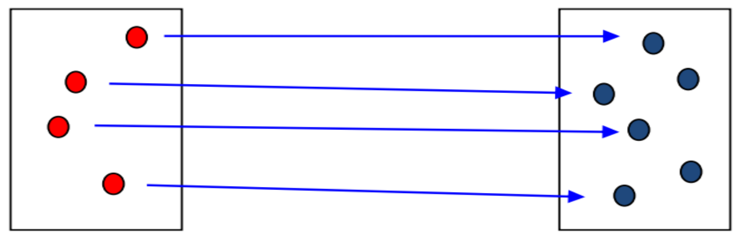


total Functions

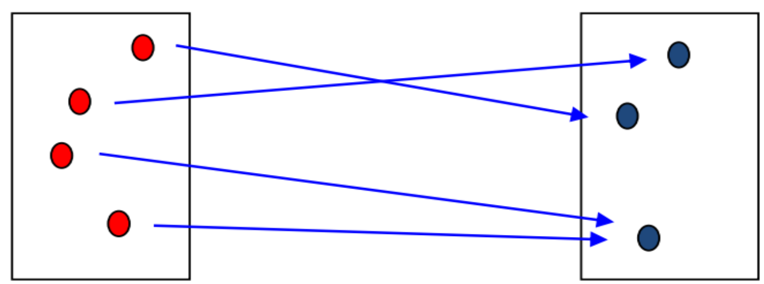


ساختارهای توابع

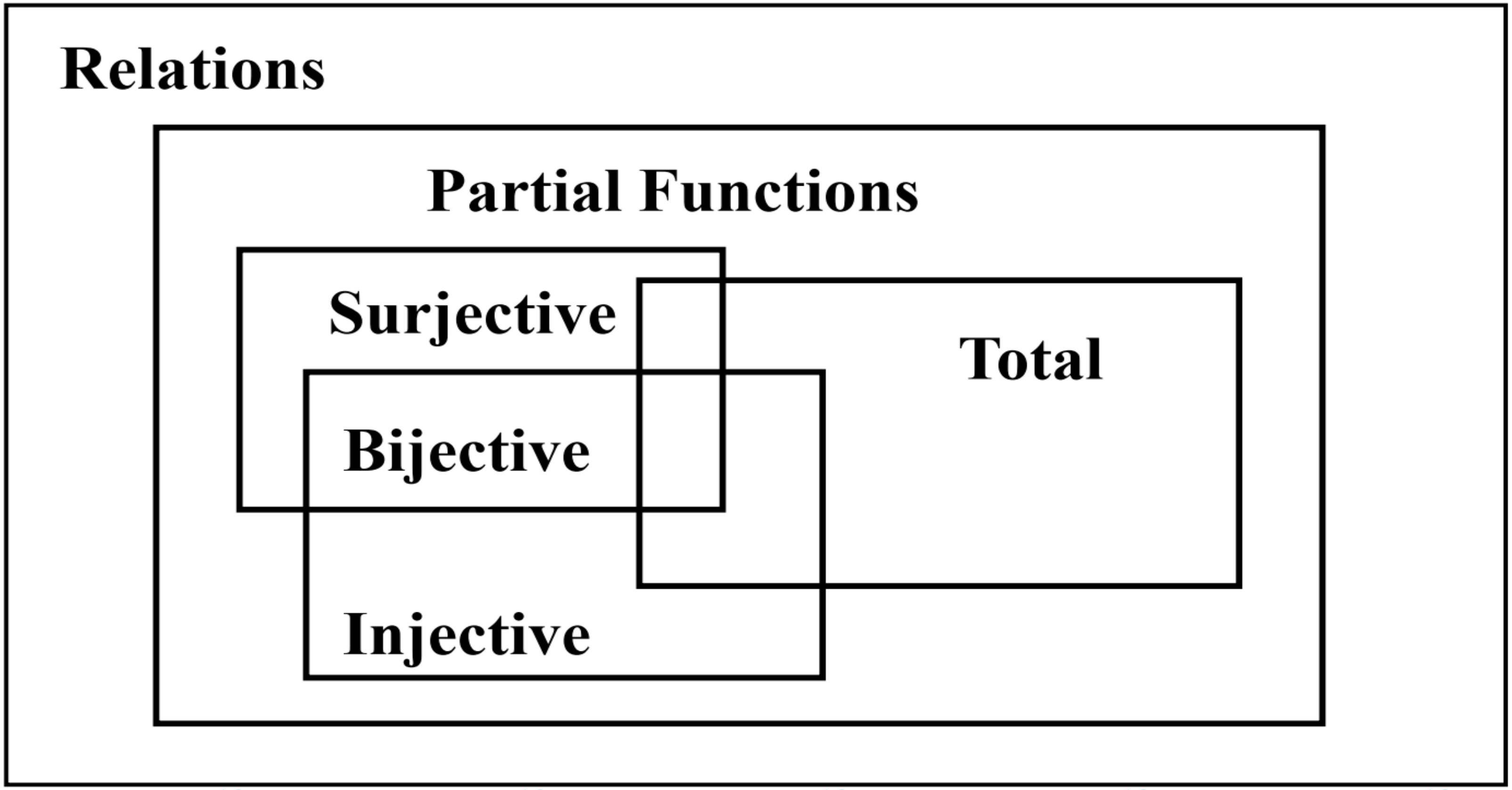
Injective Function



Surjective Function



رابطه و توابع



توابع به عنوان مجموعه

- Functions are relations and hence sets

- We can apply to them all the usual operators

  – ClassGrades = { (Todd, A), (Jane, B) }
  – #(ClassGrades ∪ { (Matt, C) }) = 3

ترکیب رابطه‌ها

- Use two relations to produce a new one – map domain of first to image of second – Given s: A x B and r: B x C then s;r : A x C

  s;r ≡ { (a,c) | (a,b) ∈ s and (b,c) ∈ r }

- For example

  – s = { (red,1), (blue,2) }
  – r = { (1,2), (2,4), (3,6) }
  – s;r = { (red,2), (blue,4) }

Not limited to binary relations

بستار انتقالی رابطه

- Intuitively, the transitive closure of a binary relation r: S x S, written r+, is what you get when you keep navigating through r until you can't go any farther.
- $$r+ \equiv r \cup (r;r) \cup (r;r;r) \cup \ldots$$
- Formally, r+ ≡ smallest transitive relation containing r

(Relation Transpose) ترانهاده رابطه

- Intuitively, the transpose of a relation r: S x T, written ~r, is what you get when you reverse all the pairs in r
- $$\sim r \equiv \{ (b,a) \mid (a,b) \in r \}$$
- For example
  - ChildOf = ~Parent
  - DescendantOf = (~Parent)+

قدردانی

- Haniel Barbosa from uiowa university
- TA: Parisa Akhbari