Relational Operations: Overriding

$r = \{(a, 1), (b, 2), (c, 3), (a, 4), (b, 5), (c, 6), (d, 1), (e, 2), (f, 3)\}$

Example: Calculate r **overridden with** $\{(a, 3), (c, 4)\}$

Hint: Decompose results to those **in** t's domain and those **not in** t's domain.

$$r \lhd t = \{(d, r') \mid (d, r') \in t \lor ((d, r') \in r \land d \notin dom(t))\}$$

(r: relation; t: relation)

$$= \{(d, r') \mid (d, r') \in t\} \cup \{(d, r') \mid (d, r') \in r \land d \notin dom(t)\}$$

Exercises: Algebraic Properties of Relational Operations

$r = \{(a, 1), (b, 2), (c, 3), (a, 4), (b, 5), (c, 6), (d, 1), (e, 2), (f, 3)\}$

Define the **image** of set s on r in terms of other relational operations.

Hint: What range of value should be included?

Define r **overridden with** set t in terms of other relational operations.

Hint: To be in t's domain or not to be in t's domain?

Functional Property

$$\text{isFunctional}(r) \Leftrightarrow$$
$$\forall s, t1, t2 \bullet$$
$$(s \in S \land t1 \in T \land t2 \in T)$$
$$\Rightarrow$$
$$((s, t1) \in r \land (s, t2) \in r \Rightarrow t1 = t2)$$

Q: Smallest relation satisfying the functional property.

Q: How to prove or disprove that a relation r is a function.

Q: Rewrite the functional property using contrapositive.

Functional Property

$$\text{isFunctional}(r) \Leftrightarrow$$
$$\forall s, t1, t2 \bullet$$
$$(s \in S \land t1 \in T \land t2 \in T)$$
$$\Rightarrow$$
$$((s, t1) \in r \land (s, t2) \in r \Rightarrow t1 = t2)$$

Q: Smallest relation satisfying the functional property.

Q: How to prove or disprove that a relation r is a function.

Q: Rewrite the functional property using contrapositive.

Partial Functions vs. Total Functions

$r \in S \nrightarrow T \Leftrightarrow (\text{ isFunctional}(r) \wedge \text{dom}(r) \subseteq S\ )$

$r \in S \rightarrow T \Leftrightarrow (\text{ isFunctional}(r) \wedge \text{dom}(r) = S\ )$

Exercise: Visualize $S \nrightarrow T$ vs. $S \longrightarrow T$

e.g., $\{\ \{(2, a), (1, b)\}, \{(2, a), (3, a), (1, b)\}\ \} \subseteq \{1, 2, 3\} \nrightarrow \{a, b\}$

e.g., $\{(2, a), (3, a), (1, b)\} \in \{1, 2, 3\} \rightarrow \{a, b\}$

e.g., $\{(2, a), (1, b)\} \notin \{1, 2, 3\} \rightarrow \{a, b\}$

e.g., $\{(2, a), (1, b), (3, a), (1, a)\} \notin \{1, 2, 3\} \rightarrow \{a, b\}$

Relational **Image** vs. Functional **Application**

A function is a relation.

$$f \in \{1, 2, 3\} ⇸ \{a, b\}$$

$$f = \{ (3, a), (1, b) \}$$

Exercises:

$f[\{3\}] =$

$f[\{1\}] =$

$f[\{2\}] =$

Modelling Decision: Relations vs. Functions

An organization has a system for keeping **track** of its employees as to where they are on the premises (e.g., ``Zone A, Floor 23''). To achieve this, each employee is issued with an active badge which, when scanned, synchronizes their current positions to a central database.

Assume the following two sets:

- *Employee* denotes the **set** of all employees working for the organization.
- *Location* denotes the **set** of all valid locations in the organization.

Is where_is $\in$ Employee <-> Location appropriate?

Is where_is $\in$ Employee $\rightarrow$ Location appropriate?

Is where_is $\in$ Employee $\nrightarrow$ Location appropriate?

Injective Functions

$\mathit{isInjective}(f)$

$\iff$

$\forall s_1, s_2, t \bullet (s_1 \in S \land s_2 \in S \land t \in T) \Rightarrow ((s_1, t) \in f \land (s_2, t) \in f \Rightarrow s_1 = s_2)$

If f is a ***partial*** ***injection***, we write: $f \in S \rightarrowtail\!\!\!\!\!\!\!+ \; T$

- e.g., $\{ \varnothing, \{(1, \mathbf{a})\}, \{(2, \mathbf{a}), (3, \mathbf{b})\} \} \subseteq \{1, 2, 3\} \rightarrowtail\!\!\!\!\!\!\!+ \; \{a, b\}$
- e.g., $\{(1, \mathbf{b}), (2, a), (3, \mathbf{b})\} \notin \{1, 2, 3\} \rightarrowtail\!\!\!\!\!\!\!+ \; \{a, b\}$
- e.g., $\{(1, \mathbf{b}), (3, \mathbf{b})\} \notin \{1, 2, 3\} \rightarrowtail\!\!\!\!\!\!\!+ \; \{a, b\}$

If f is a ***total*** ***injection***, we write: $f \in S \rightarrowtail T$

- e.g., $\{1, 2, 3\} \rightarrowtail \{a, b\} = \varnothing$
- e.g., $\{(2, d), (1, a), (3, c)\} \in \{1, 2, 3\} \rightarrowtail \{a, b, c, d\}$
- e.g., $\{(\mathbf{2}, d), (\mathbf{1}, c)\} \notin \{1, 2, 3\} \rightarrowtail \{a, b, c, d\}$
- e.g., $\{(2, \mathbf{d}), (1, c), (3, \mathbf{d})\} \notin \{1, 2, 3\} \rightarrowtail \{a, b, c, d\}$

Surjective Functions

$$\texttt{isSurjective}(f) \iff \underline{\text{ran}}(f) = \underline{T}$$

If f is a ***partial surjection***, we write: $\boxed{f \in S \twoheadrightarrow\!\!\!\!\!\!+\; T}$

- e.g., $\{ \{(1,\mathbf{b}),(2,\mathbf{a})\}, \{(1,\mathbf{b}),(2,\mathbf{a}),(3,\mathbf{b})\} \} \subseteq \{1,2,3\} \mathrel{+\!\!\!\twoheadrightarrow} \{a,b\}$
- e.g., $\{(2,\mathbf{a}),(1,\mathbf{a}),(3,\mathbf{a})\} \notin \{1,2,3\} \mathrel{+\!\!\!\twoheadrightarrow} \{a,b\}$
- e.g., $\{(2,\mathbf{b}),(1,\mathbf{b})\} \notin \{1,2,3\} \mathrel{+\!\!\!\twoheadrightarrow} \{a,b\}$

If f is a ***total surjection***, we write: $\boxed{f \in S \twoheadrightarrow T}$

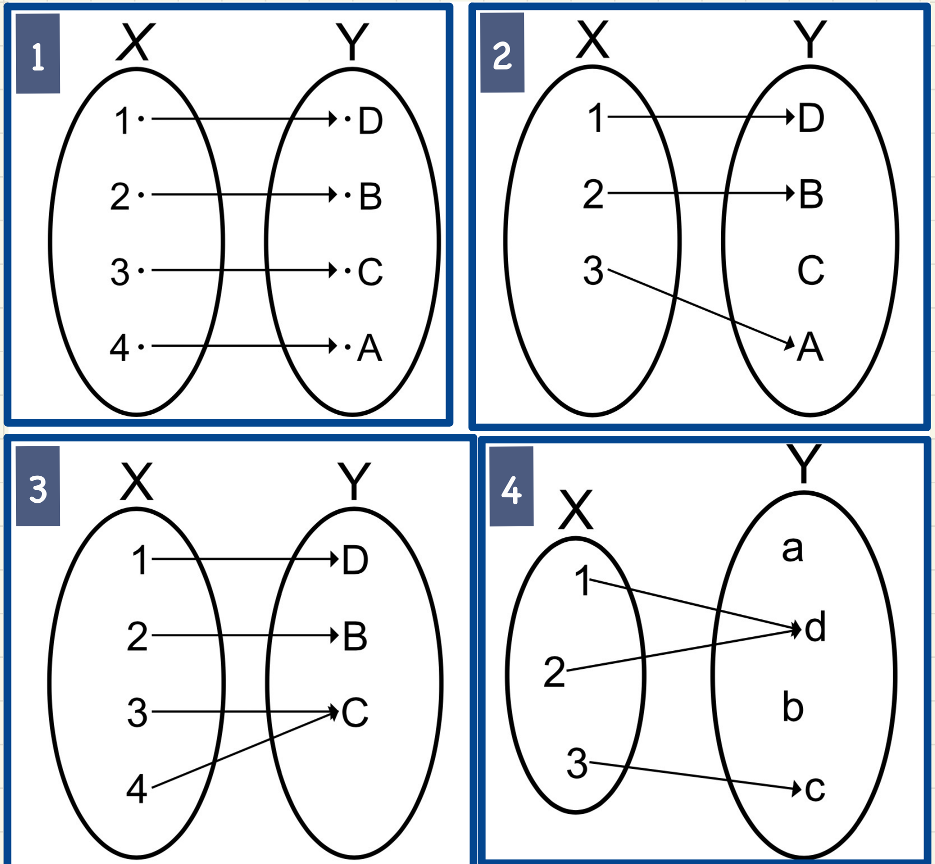
- e.g., $\{ \{(2,a),(1,b),(3,a)\}, \{(2,b),(1,a),(3,b)\} \} \subseteq \{1,2,3\} \twoheadrightarrow \{a,b\}$
- e.g., $\{(\mathbf{2},a),(\mathbf{3},b)\} \notin \{1,2,3\} \twoheadrightarrow \{a,b\}$
- e.g., $\{(2,\mathbf{a}),(3,\mathbf{a}),(1,\mathbf{a})\} \notin \{1,2,3\} \twoheadrightarrow \{a,b\}$

Bijective Functions

f is ***bijective***/***a bijection***/*one-to-one correspondence* if f is ***total***, ***injective***, and ***surjective***.

- e.g., $\{1,2,3\} \rightarrowtail\!\!\!\!\twoheadrightarrow \{a,b\} = \emptyset$
- e.g., $\{\ \{(1,a),(2,b),(3,c)\},\{(2,a),(3,b),(1,c)\}\ \} \subseteq \{1,2,3\} \rightarrowtail\!\!\!\!\twoheadrightarrow \{a,b,c\}$
- e.g., $\{(\mathbf{2},b),(\mathbf{3},c),(\mathbf{4},a)\} \notin \{1,2,3,4\} \rightarrowtail\!\!\!\!\twoheadrightarrow \{a,b,c\}$
- e.g., $\{(1,\mathbf{a}),(2,b),(3,c),(4,\mathbf{a})\} \notin \{1,2,3,4\} \rightarrowtail\!\!\!\!\twoheadrightarrow \{a,b,c\}$
- e.g., $\{(1,\mathbf{a}),(2,\mathbf{c})\} \notin \{1,2\} \rightarrowtail\!\!\!\!\twoheadrightarrow \{a,b,c\}$

Exercise



	1	2	3	4
partial				
total				
injection				
surjection				
bijection				

Formalizing Arrays as Functions

```
String[] names = {"alan", "mark", "tom"};
```