Functions

- Functions form a special class of relations that satisfy additional requirement: any element of the source set can be related to no more than 1 element of the target
- Functionality requirement mathematically:

$$\forall x, y, z.\ \ (x \mapsto y) \in R \ \wedge\ (x \mapsto z) \in R \ \Rightarrow\ \ y = z$$

- Any operation applicable to a relation or a set is also applicable to a function. For example, we can talk about the domain and the range of a function or a function as a set of pairs
- If f is a function, then $f(x)$ is the result of the function f for the argument x

Total and partial functions

- Functions are called *total* if their domain is the whole source set
  Syntax: $f \in S \rightarrow T$ (or ascii $f : S \text{ --> } T$)
  where $dom(f) = S$ and $ran(f) \subseteq T$
- Example: the camera model relation is actually a total function
  $cmodels \in CMODEL \rightarrow CAMERA$
  (any camera model identifier is uniquely associated with its brand)
- Functions are called *partial* if their domain is a subset of the source set
  Syntax: $f \in S \nrightarrow T$ (or ascii $f : S \text{ +-> } T$)
  where $dom(f) \subseteq S$ and $ran(f) \subseteq T$
- Example: room reservation relation is actually a partial function
  $reserved \in ROOM \nrightarrow CUSTOMER$
  (each reserved room has the unique customer that served it, however, not all rooms must be reserved)

Arrays

- An array is a named, indexed collection of values of a given type.
- The array values can be accessed (read and updated) by using appropriate indexes.
- If we use $1..n$ (for some $n \in \mathbb{N}$) as our index set, then an array (containing elements of type S) can be modelled as a function from $1..n$ to S.
- In fact, any set can be used as the index set for arrays. Therefore, arrays can be usually modelled as total functions from S (index set) to T (the type of array values).

Functional (array) assignment

- The notation used to describe machine actions (i.e., assignments in the machine events) allows us to directly assign values to indexed elements of arrays:

$$a(i) := E$$

- This is just syntactic sugaring for the following assignment:

$$a := a \lhd \{(i \mapsto E)\}$$

- The assignment also works if a is modelled as a partial function. However, if we want to check/read values from such an array, we have to ensure/prove (by using the event guards and/or machine invariants) that the used index belongs to the function domain, i.e., $i \in dom(a)$

- $reserved(r) := FALSE$ **OK!**
- $nItems(j) := nItems(i) + 1$ **only if** $i \in dom(nItems)$