

## Object Machines

<i>Values</i>	<i>Objects</i>
Integer, real, char, Boolean	All other data types
	<b>Identity</b>  <b>State</b> $\tau$ (a tuple $\tau$ of primitive values and object references ex. ref<X>) R1: No shared states  <b>Behaviour</b> (responses to messages with operations) (a) <i>Signatures</i> (b) <i>Axioms</i> (c) <i>Protocol machine</i>

State of an object:

$\tau.\alpha$

$\tau.\alpha \leftarrow v$

$\tau = \mu \cup \kappa$ ,  $\mu$ : memory(primitive computations),  
 $\kappa$ : collaborators(computations involve communication with other objects).

$\mu \cap \kappa = \emptyset$

**Example:**

*Array*: collaborator of a stack

Representation:  $\text{Array}\langle\text{Value}\rangle$

Behaviour = (a)+(b)+(c):*operations*: put, get, size

(a) *Signatures*: ...

(b) *Axioms*: ...

Protocol machine: (c) *protocol states* of the Array

*Axioms*

Testing: by using Chow's method.

**Example:**

*The bounded Stack*

Storage Locations: Loaded

Full

Representation:  $\text{Stack}\langle\text{Value}\rangle$

Behaviour = (a)+(b)+(c):*operations*: push, pop, top, empty, full, size, capacity

(a) *Signatures*: ...

(b) *Axioms*: ...

Protocol machine: (c) *protocol states* of the Array

*Axioms*

Testing: by using Chow's method.