

**CSC 4504 : *Langages formels et applications***

**(Event-B)**

**J Paul Gibson, A207**

`paul.gibson@it-sudparis.eu`

<http://www-public.it-sudparis.eu/~gibson/Teaching/CSC4504/>

**Arrays and Algorithms**

<http://www-public.it-sudparis.eu/~gibson/Teaching/CSC4504/Arrays&Algorithms.pdf>

# Finding the maximum element in an array of NATs

## CONTEXT

ArrayOfNATs\_ctx

## CONSTANTS

Size

ArrayOfSizeNats

Maximum

## AXIOMS

axm1 : Size  $\in$  N

axm2 : ArrayOfSizeNats = 1.. Size  $\rightarrow$  N

axm3 : Maximum  $\in$  ArrayOfSizeNats  $\rightarrow$  N

axm4 :  $\forall$  values, maxi  $\cdot$  values  $\in$  ArrayOfSizeNats  $\wedge$  maxi  $\in$  N  $\Rightarrow$   
((values  $\mapsto$  maxi)  $\in$  Maximum  $\Leftrightarrow$   
( $\forall$  i  $\cdot$  i  $\in$  1 .. Size  $\Rightarrow$  values(i)  $\leq$  maxi) )

axm5 :  $\forall$  values, maxi  $\cdot$  values  $\in$  ArrayOfSizeNats  $\wedge$  maxi  $\in$  N  $\Rightarrow$   
( Maximum (values) = maxi  $\Leftrightarrow$   
( $\forall$  i  $\cdot$  i  $\in$  1 .. Size  $\Rightarrow$  values(i)  $\leq$  maxi) )

END

Type this new context into RODIN

## Finding the maximum element in an array of NATs

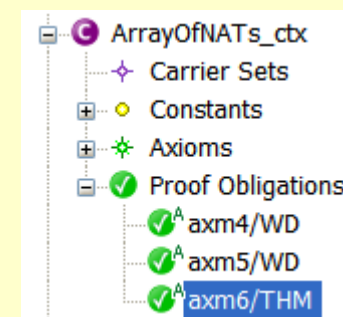
QUESTION: **Validate** that this formal specification is a complete and consistent specification of a maximum of an array of natural numbers

HINT: Add some validating theorems and check that they are true

NOTE: The Size is a generic parameter, so you may need to write theorems in the following form:

```
axm6 : Size = 1 ⇒ { 1 ↦ 1 } ∈ ArrayOfSizeNats theorem //
```

```
axm6 : Size = 1 ⇒ { 1 ↦ 1 } ∈ ArrayOfSizeNats
```



# VOTE TRANSFER/TRANSFORMATION IN E-VOTING SYSTEMS

In voting machines there have been many reports of votes being lost and/or "changed" as they move through the voting system.

In particular, votes must move from a machine interface to a central counting location.

This move can prove to be problematic.

**Experiment:** Can formal methods help to reduce the risk of programmers avoiding these errors?

# VOTE TRANSFER/TRANSFORMATION IN E-VOTING SYSTEMS

Consider the following specific example:

In Irish elections voters record a preference for candidates. For example, a typical ballot is recorded at the interface as:

$[0,2,0,0,1]$  and this represents 5 candidates where the first preference is for candidate 5 and the second preference is for candidate 2.

However, this vote is represented differently in the count module:

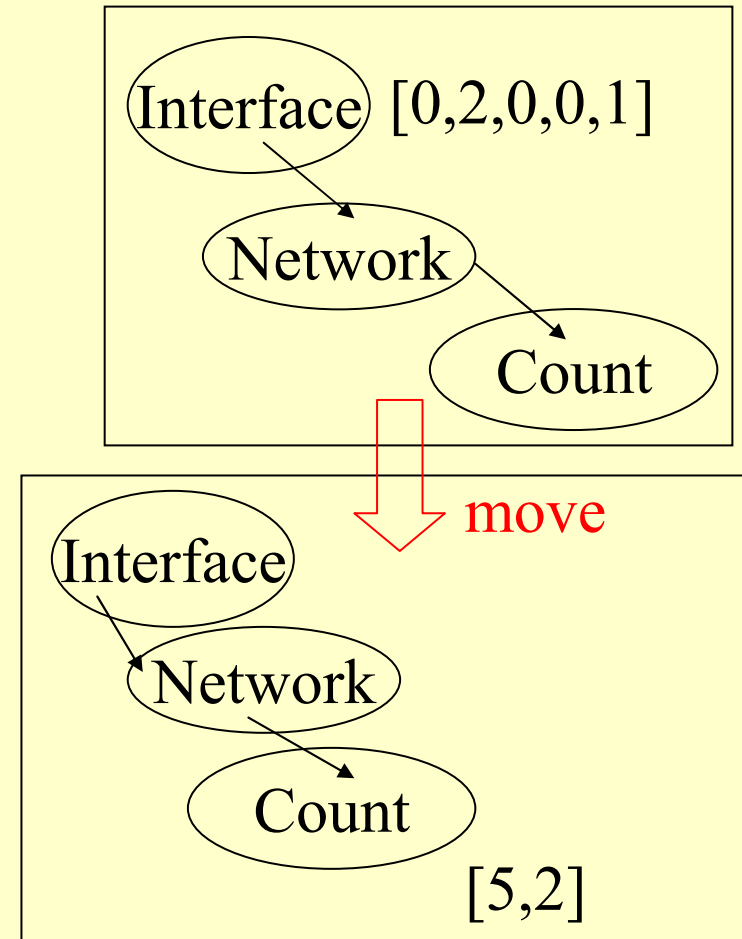
$[5,2]$

# VOTE TRANSFER/TRANSFORMATION IN E-VOTING SYSTEMS

**TASK 1:** In Java or C, implement the code that can move ballots from interface to count module, and from count module to interface.

The code must follow the following design constraint that models the use of a network to connect the interface component to the count component.

There is no direct communication between the interface and the count. Instead, the interface can push a pair of integer values onto the "network" and the count can pop the pair of integer values from the "network". The "network" is limited to storing a single pair of integer values.



# VOTE TRANSFER/TRANSFORMATION IN E-VOTING SYSTEMS

## TASK 2:

Once you are happy that your code is correct for task 1, consider the following additional requirement:

*The network link may be unreliable and may lose data that is currently stored on the "network". To address this problem, your engineers suggest that you may wish to add a second network link that allows communication in the opposite direction of the first, i.e from count module to vote interface. This second link is (like the first) restricted to only pushing and popping pairs of integer values (and with the same single pair limit for storing the values). Your engineers also tell you that each network will automatically detect multiple occurrences of loss of data (>1 time) and subsequently guarantee that this network will never lose data again. In other words, each network connection will lose at most 2 pieces of data during the election.*

**PROBLEM:** Implement code for moving ballots from the interface to the count module - using only the 2 network connections - that guarantees that no ballot is lost.

# VOTE TRANSFER/TRANSFORMATION IN E-VOTING SYSTEMS

**CHOOSE ONE OF THE FOLLOWING DEVELOPMENT PROCESSES:**

- Code and Test
- Specify in Event-B then Code and Test
- Design (in language(s) of your choice) then Code and Test

You must use the same process for each task

At the end of the experiment email only your current code and tests to

[Paul.gibson@it-sudparis.eu](mailto:Paul.gibson@it-sudparis.eu) ; and specify which process you followed in the subject field of the email

Please email me all the versions you develop in 1 zipped directory (containing separate subdirectories labelled v1,v2, ...)