

# An Event-B Specification of Vectors

---

This project tests code generation for vectors.

---

<b>1</b>	<b>CONTEXT Context</b>	<b>2</b>
1.1	hi hii index lo loo . . . . .	2
<b>2</b>	<b>MACHINE Test</b>	<b>3</b>
2.1	bytes bytes_size . . . . .	3
2.2	cut(new_size) . . . . .	3
2.3	split(left_size new_size out_msg) . . . . .	3
<b>3</b>	<b>REFINEMENT Test1</b>	<b>4</b>
3.1	. . . . .	4
3.2	split extends split . . . . .	4
<b>4</b>	<b>MACHINE Vectors</b>	<b>5</b>
4.1	bytes bytes_size heights . . . . .	5
4.2	setHeight(at h) . . . . .	5
4.3	setHeights(hs) . . . . .	5
4.4	findHeight(h out_i) . . . . .	5
4.5	addByte(b) . . . . .	6
4.6	cut(new_size) . . . . .	6

## CONSTANTS

1.1

lo  
hi  
index  
loo  
hii

## AXIOMS

axm1: lo = 1  
axm2: hi = 10  
axm3: index = lo..hi  
axm4: loo  $\in \mathbb{Z}$   
axm5: hii  $\in \mathbb{Z}$   
axm6: hii > loo

## THEOREM

thm1: card(index) = max(index)  
thm2: lo = min(index)  
thm3: hi = max(index)  
thm4: 5 = card(5..9)  
thm5: 9 = max(5..9)  
thm6: hii = max(loo..hii)

END

## VARIABLES

2.1

*bytes*  
*bytes\_size*

## INVARIANTS

*inv1:*  $bytes \in 1..bytes\_size \rightarrow 0..255$   
*inv2:*  $bytes\_size = \text{card}(\text{dom}(bytes))$

## EVENT INITIALISATION

## THEN

*init1:*  $bytes := 1..10 \times \{1\}$   
*init2:*  $bytes\_size := 10$

## END

## EVENT cut

2.2

## ANY

*new\_size*

## WHERE

*grd1:*  $bytes\_size > 1$   
*grd2:*  $new\_size = bytes\_size - 1$   
*grd3:*  $new\_size \in \mathbb{N}_1$

## THEN

*act1:*  $bytes\_size := new\_size$   
*act2:*  $bytes := \{x \cdot x \in 1..new\_size \mid x \mapsto bytes(x + 1)\}$

## END

## EVENT split

2.3

## ANY

*new\_size*  
*left\_size*  
*out\_msg*

## WHERE

*grd1:*  $new\_size \in \mathbb{N}_1$   
*grd2:*  $left\_size \in \mathbb{N}_1$   
*grd3:*  $bytes\_size > left\_size$   
*grd4:*  $new\_size = bytes\_size - left\_size$   
*grd5:*  $out\_msg = 1..new\_size \triangleleft bytes$

## THEN

*act1:*  $bytes\_size := new\_size$   
*act2:*  $bytes := \{x \cdot x \in 1..new\_size \mid x \mapsto bytes(x + left\_size)\}$

## END

REFINEMENT <code>Test1</code>	1 	<div>3</div>
<hr/>		
REFINES <code>Test</code>		
VARIABLES		<div>3.1</div>
EVENT <code>split</code>		<div>3.2</div>
EXTENDS <code>split</code>		
WHERE		
<code>grd1_1:</code> <code>2</code> $\in \text{dom}(\text{bytes})$		
<code>grd1_2:</code> $\text{left\_size} = \text{bytes}(\text{1})$		
END		

## VARIABLES

4.1

*heights*     A fixed size vector of heights.  
*bytes\_size*   Size of the bytes vector  
*bytes*        A vector of bytes

## INVARIANTS

**inv\_he:**     $heights \in (1..100) \rightarrow \mathbb{N}$   
**inv\_bs:**     $bytes\_size \in \mathbb{N}$   
**inv\_by:**     $bytes \in (1..bytes\_size) \rightarrow 0..255$

EVENT **INITIALISATION**

## THEN

**init\_he:**     $heights := 1..100 \times \{0\}$   
**init\_ws:**     $bytes\_size := 0$   
**init\_we:**     $bytes := \emptyset$

## END

EVENT **setHeight**

4.2

## ANY

*at*  
*h*

## WHERE

**grd\_p:**     $at \in \text{dom}(heights)$   
**grd\_h:**     $h \in \mathbb{N}$

## THEN

**act\_1:**     $heights(at) := h$

## END

EVENT **setHeights**

4.3

## ANY

*hs*

## WHERE

**grd\_hs:**     $hs \in (1..100) \rightarrow \mathbb{N}$

## THEN

**act\_1:**     $heights := hs$

## END

EVENT **findHeight**

4.4

## ANY

*h*  
*out\_i*

## WHERE

**grd1:**     $h \in \mathbb{N}$   
**grd3:**     $out\_i \in \text{dom}(heights)$   
**grd4:**     $\exists x. x \in \text{dom}(heights) \wedge height(x) = h \wedge out\_i = x$

## END

EVENT **addByte**

4.5

ANY

$b$

WHERE

**grd** $b$ :  $b \in 0..255$

THEN

**act** $_1$ :  $bytes := bytes \cup \{bytes\_size + 1 \mapsto b\}$

**act** $_2$ :  $bytes\_size := bytes\_size + 1$

END

EVENT **cut**

4.6

ANY

$new\_size$

WHERE

**grd** $_1$ :  $bytes\_size > 1$

**grd** $_2$ :  $new\_size \in \mathbb{N}_1$

**grd** $_3$ :  $new\_size = bytes\_size - 1$

THEN

**act** $_1$ :  $bytes\_size := new\_size$

**act** $_2$ :  $bytes := \{x \cdot x \in 1..new\_size \mid x \mapsto bytes(x + 1)\}$

END

addByte, 6

bytes, 3, 5

bytes\_size, 3, 5

Context, 2

cut, 3, 6

findHeight, 5

heights, 5

INITIALISATION, 3, 5

setHeight, 5

setHeights, 5

split, 3, 4

Test, 3, 4

Test1, 4

Vectors, 5