

1) The figure below shows the test feature `is_negative` at the call to `compare_functions`. In the call, I pass "agent n.is_negative" as the argument, where `n` is a JJ_BIG_NATURAL_8.

The screenshot shows the Eclipse IDE with the following components:

- Source Editor:** Displays the source code for the feature `is_negative`. The code includes a loop that calls `compare_functions (agent n.is_negative, True, fn, ARRAY [ANY] <<>>)`.
- Right Panel:** Shows a table of feature instances. The row for `is_negative` in class `BIG_NATURAL_8_TESTS` is highlighted with a red circle.
- Bottom Panel:** Shows the execution state for the feature `(BIG_NATURAL_8_TESTS).is_negative`. The `Locals` table shows:

Name	Value	Type	Address
fn	is_negative	STRING_8	0x7FD06E380D50
i	1	INTEGER_32	
n	-595901053	JJ_BIG_NATURAL_8	0x7FD06E380D88

However, inside feature `compare_functions`, the argument is not a JJ_BIG_NATURAL_8; it is a BIG_NATURAL_8_TEST, which is the type of the enclosing class at the call to `compare_functions`. Hence it fails on the precondition. ?????

The screenshot shows the Eclipse IDE with the following components:

- Source Editor:** Displays the source code for the feature `compare_functions`. The code includes a precondition check: `require -- from BIG_NATURAL_TESTS target closed: attached {like number_anchor} a_routine.target other target closed: attached {ROUTINE} a_expected as r implies attached {like number_anchor} r.target`.
- Right Panel:** Shows a table of feature instances. The row for `compare_func...` in class `BIG_NATURAL_8_TESTS` is highlighted with a red circle.
- Bottom Panel:** Shows the execution state for the feature `(BIG_NATURAL_8_TESTS).compare_functions`. The `Arguments` table shows:

Name	Value	Type	Ad
a_routine	<0x7FD06E380CF0>	PREDICATE [ITUPLE]	0x7
Agent	is_negative		
closed_op...	<0x7FD06E380DF8>	TUPLE [BIG_NATURAL_8_TESTS, IJ_BIG_NATURAL_8]	0x7

2) Feature `set_with_array` was giving strange results in the debugger. In the debugger the three circled values in the picture below were all different; they should have been the same, as they are now. Argument `a_array` came in with certain values but the value assigned to `d` was different; when that value was placed into Current, it was again a different value. Then, I noticed that the local variable `place` was never used, and I had no idea why I had put that there, so I deleted that line. Now the procedure works! I then I put the unused local back in (i.e. undo edit) and, wow, the procedure STILL works as expected; I now cannot reproduce the error. ???

The screenshot shows the source code for the feature `set_with_array` and a debugger's variable table. The code includes a loop that iterates over an array and extends it into the current context. The variable table below shows the state of various variables, with three values circled in red: 242, 242, and 242.

Code Snippet:

```

set_with_array (a_array: ARRAY [like word])
-- Set the words from `a_array`, where the array holds
-- intended words with high-order words first.
-- (from JJ_BIG_NATURAL)
require -- from JJ_BIG_NATURAL
array_exists: a_array /= Void
array_not_empty: not a_array.is_empty

local
i: INTEGER_32
d: like word
place: JJ_BIG_NATURAL_8

do
wipe_out
from
i := a_array.count
until
i < 1
loop
d := a_array [i]
extend (a_array [i])
i := i - 1
end
end

```

Variable Table:

Name	Value	Type	Addr	Expression
count	1			a_array.index
capacity	10			...
0	242	NATURAL_8		
index	0	INTEGER_32		
is_negative	False	BOOLEAN		
is_unstable	False	BOOLEAN		
object_comp...	False	BOOLEAN		
a_array	<0x7F846AB80D38>	ARRAY [NATURAL_8]	0x7F...	
area	count=4, capacity=4	SPECIAL [NATURAL_8]	0x7F...	
count	4			
capacity	4			
0	187	NATURAL_8	0x7F...	
1	133	NATURAL_8	0x7F...	
2	35	NATURAL_8	0x7F...	
3	242	NATURAL_8	0x7F...	
lower	1	INTEGER_32		
object_co...	False	BOOLEAN		
upper	4	INTEGER_32		
d	242	NATURAL_8	0x7F...	
i	4	INTEGER_32		
place	Void	JJ_BIG_NATURAL	Void	

Callout 1 (Top): What is this? After deleting this unused local, this procedure seems to work fine. ??? But after replacing the line, the error is no longer happening. ???

Callout 2 (Bottom): Before deleting `place`: on entry, a_array [4] was 242. After assignment, `d` became 168. Extending into Current added some other value [don't remember what that value was, because it works now.]