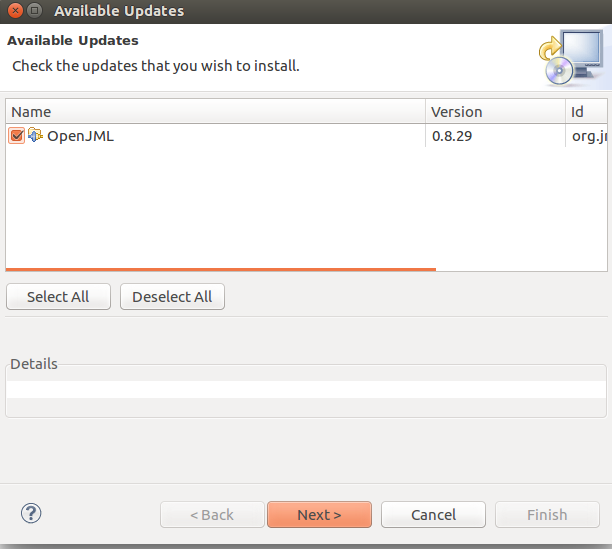
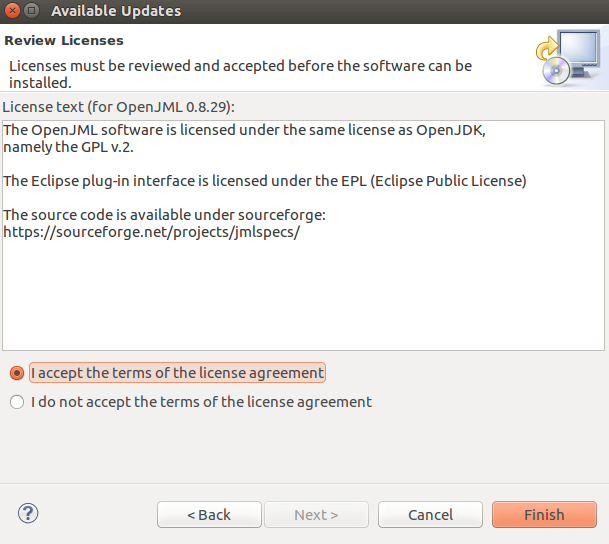
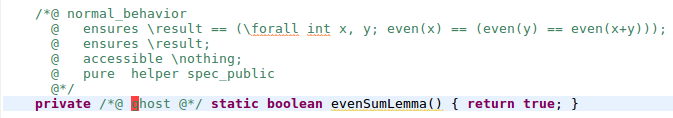
# OpenJML RAC Update

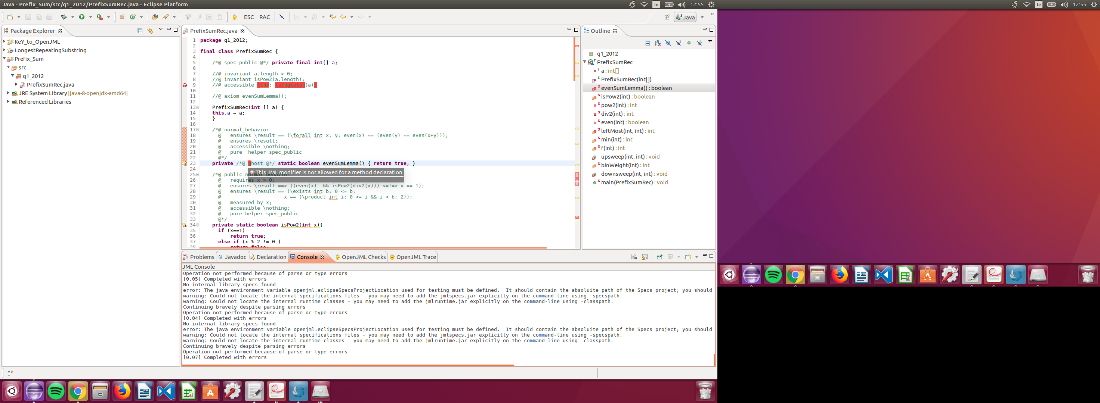




Note: the ability to use ghost methods would allow the use of Lemmas as done in KeY and Why3



Returns error: This JML modifier is not allowed for method declarations



# PrefixSumArray

Type-Checking

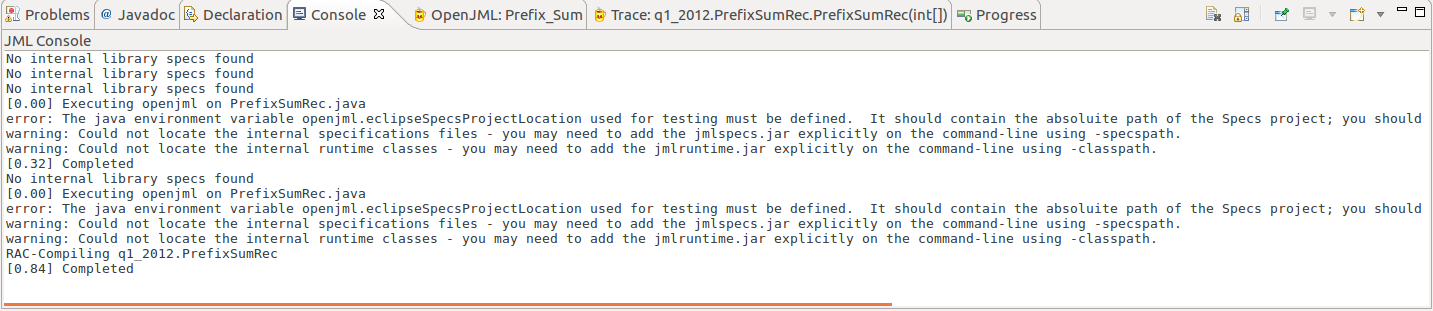
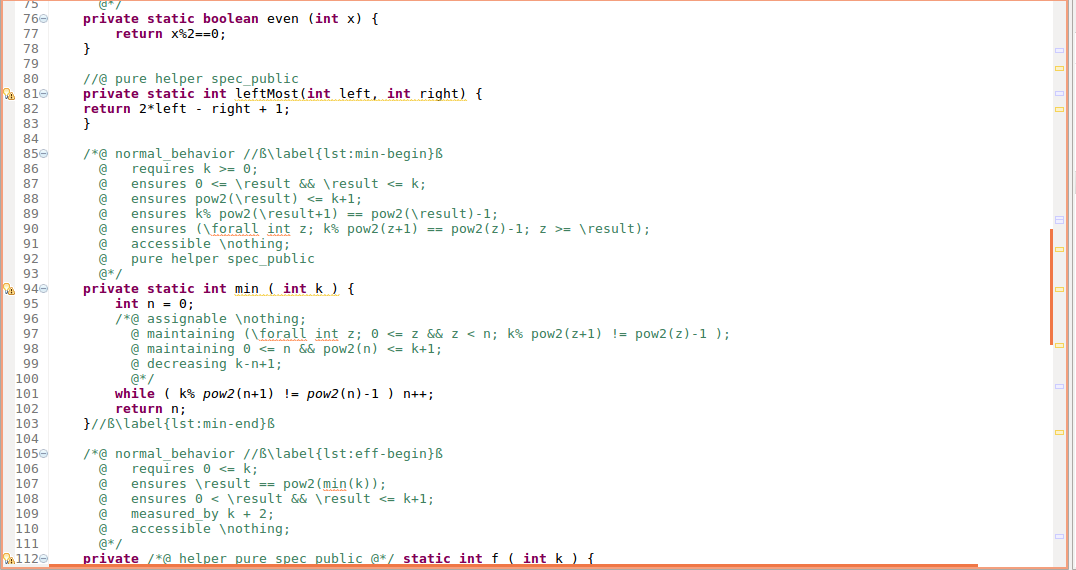
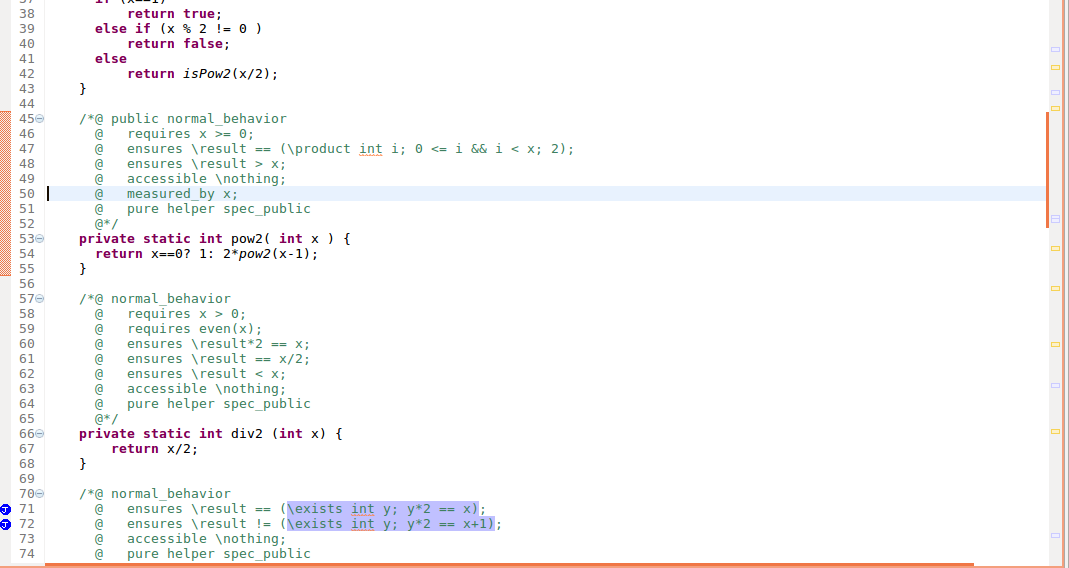
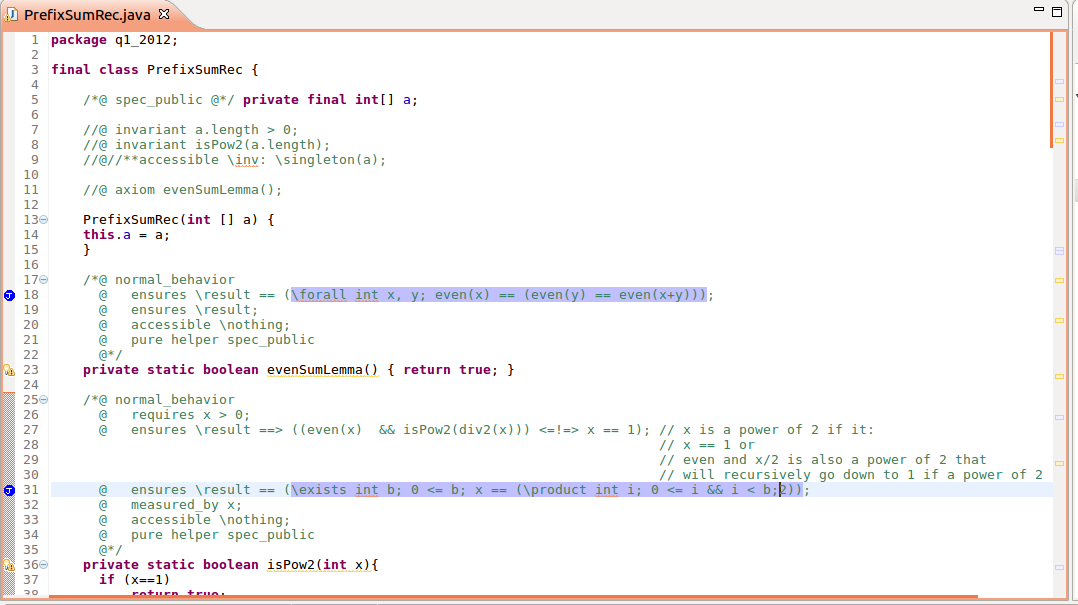
13:37: Partially converted KeY to OpenJML

* Multiple //@ not working so replaced with //@//\*\*
* \singleton
* \infinite\_union

## RAC

15:36 – RAC performed

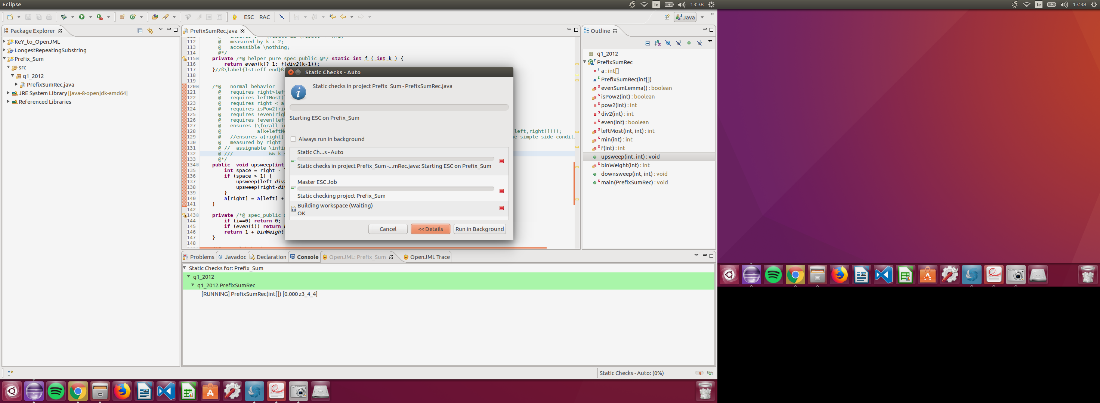
* Blue Icon:
  + Runtime assertion checking is not implemented for this type or number of declarations in a qualified expression.
  + Counterexample: No proof information available
* No Errors found



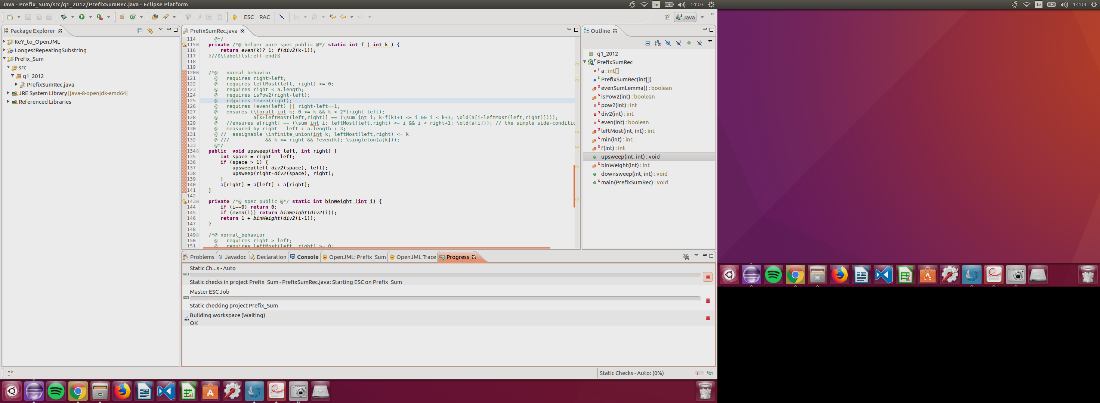
## ESC

### Eclipse

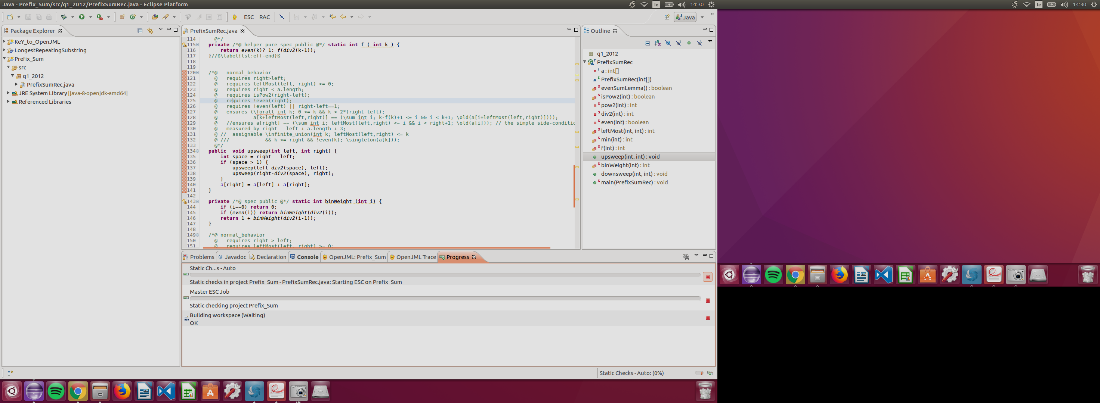
1. 13:38 - First Attempt at RAC/ESC on PrefixSumArray in Eclipse



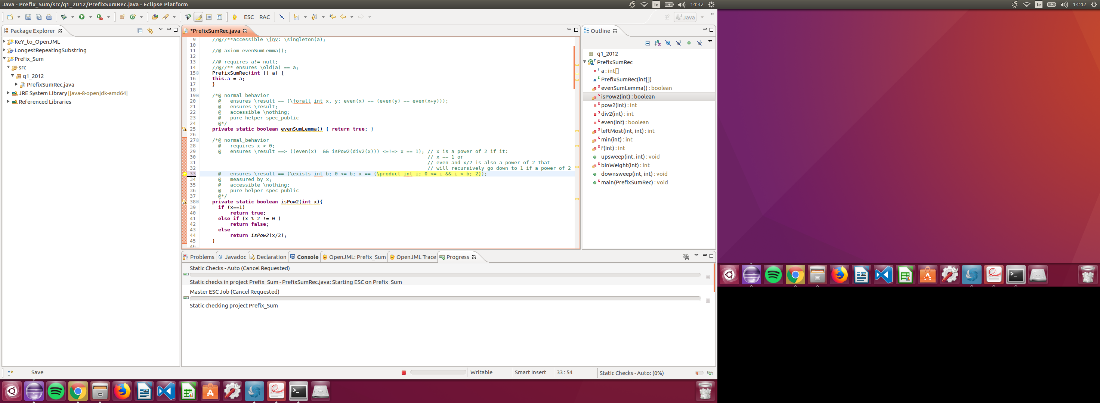
14:03 - 0% progress



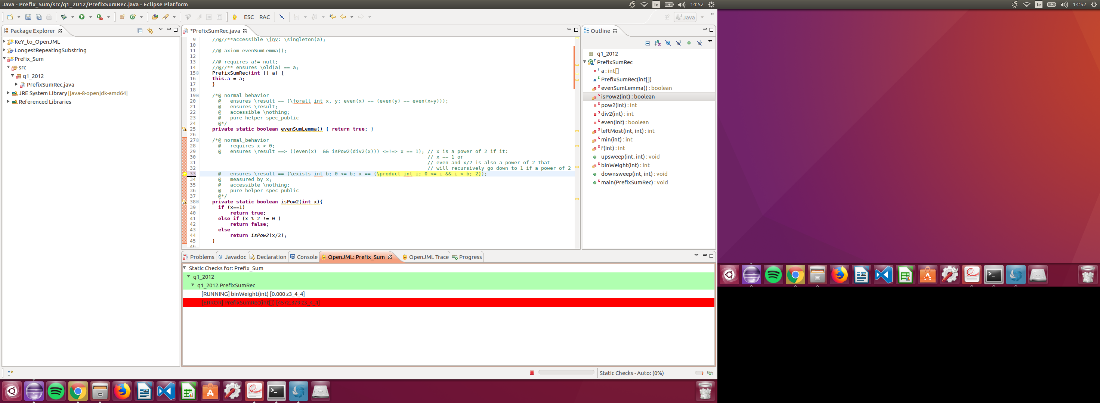
14:29 – 0% Progress (Eclipse ESC cancelled)



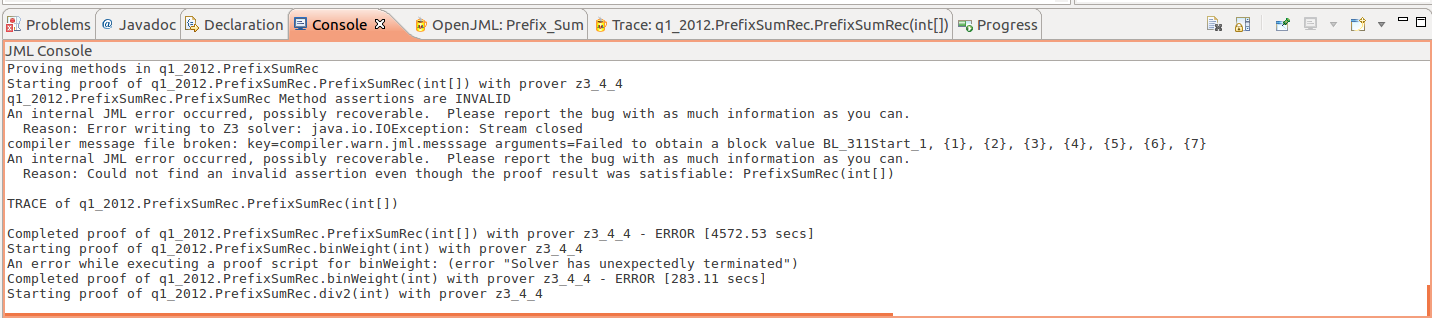
14:42 Eclipse hung when cancelling operation



14:52 – Eclipse won't exit and continues to try ESC with a result after z3 process was forcefully stopped



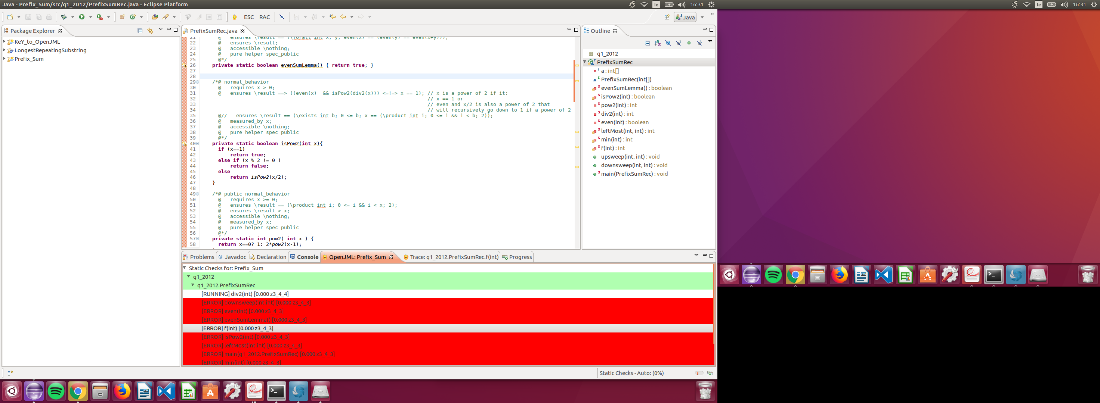
14:57 – Error appears to be with binWeight method



1. Second ESC attempt

16:32 – prover error

* Eclipse not recognizing prover executable despite setting it in preferences



* Prover started working
* Note: A measured by clause can be used in a termination argument for a recursive specification
  + A pure method or constructor must also be provably terminating.[(19)](http://www.dc.fi.udc.es/ai/tp/practica/jml/JML/docs/prelimdesign/prelimdesign/prelimdesign_foot.html#FOOT19) Recursion is permitted, both in the implementation of pure methods and the data structures they manipulate, and in the specifications of pure methods. When recursion is used in a specification, the proof of well-formedness for the specification involves the use of JML's measured\_by clause.
    - *Dc.fi.udc.es. (2018). Preliminary Design of JML - 2. Class and Interface Specifications. [online] Available at:* [*http://www.dc.fi.udc.es/ai/tp/practica/jml/JML/docs/prelimdesign/prelimdesign/prelimdesign\_2.html*](http://www.dc.fi.udc.es/ai/tp/practica/jml/JML/docs/prelimdesign/prelimdesign/prelimdesign_2.html) *[Accessed 26 Apr. 2018].*



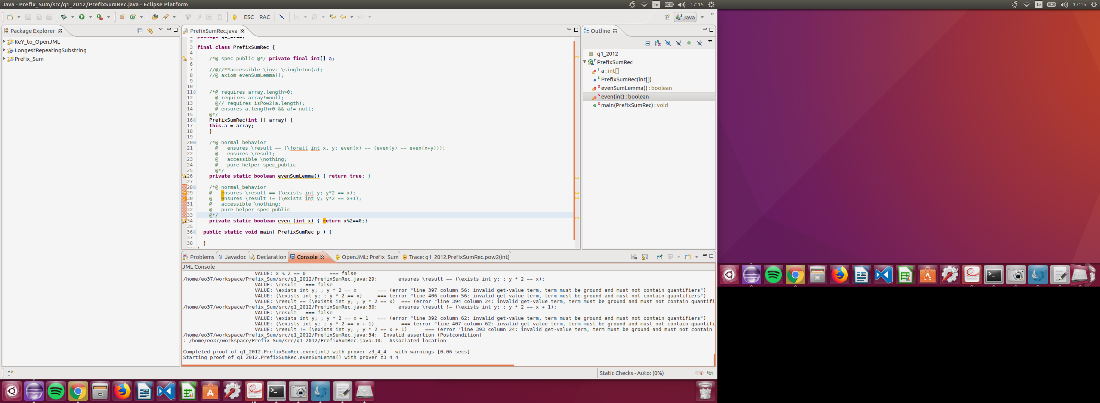
17:02 – Too many errors found to work through, we will have to verify each method individually as they all call each other in their own specifications

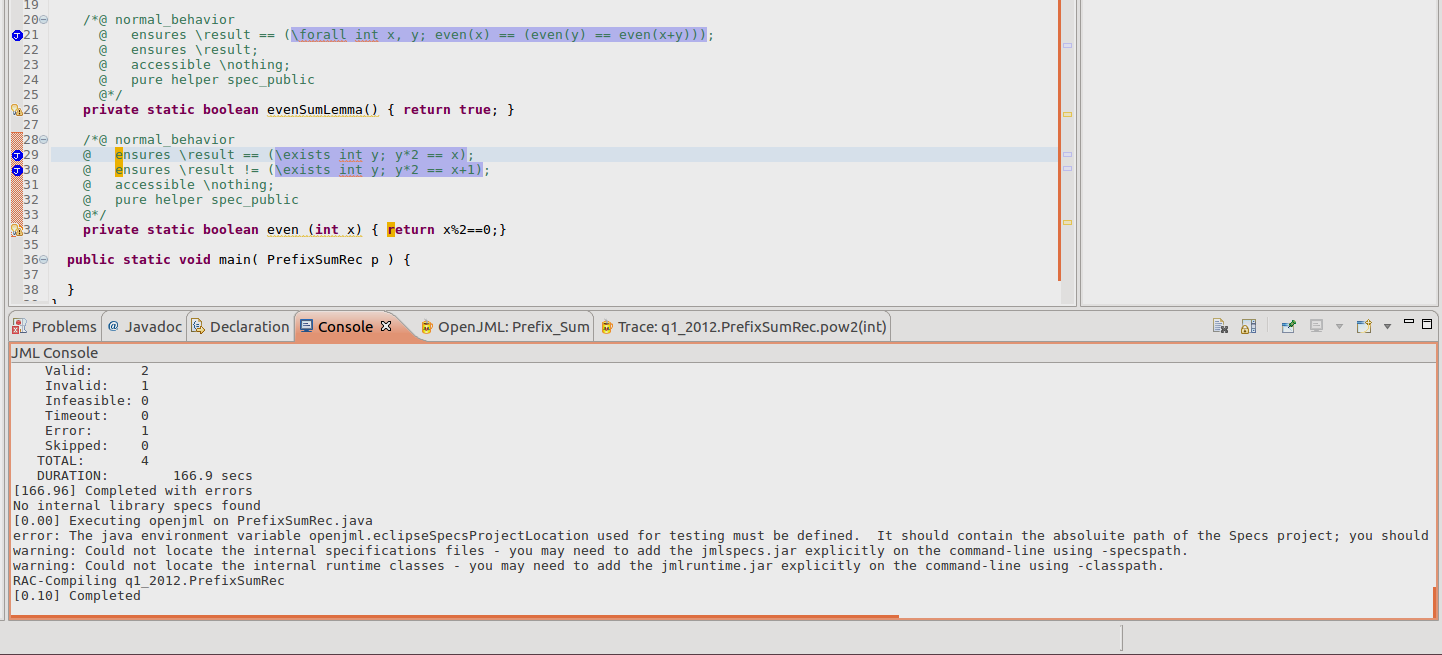
17:15 – Starting with evenSumLemma method which requires the even method in its spec

* Error verifiying the even method

z3 does not support evaluation of quantified formulas

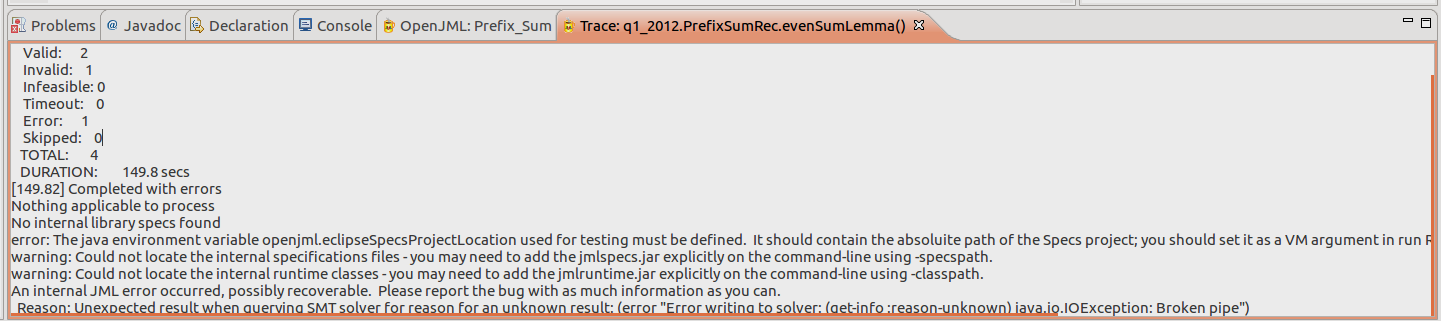
* + OpenJML cannot evaluate this separately to other parts of the spec. Major issue





17:38 – Removed two quantifed formulas from even method

* even method now passes
* Internal error with io pipe for evenSumLemma method
* Line 21: @ ensures \result == (\forall int x, y; even(x) == (even(y) == even(x+y))); is causing the prover to timeout



20:12 – Added in pow2, div2 and isPow2 methods

* div2 specification is invalid
  + div2 method: @ ensures \result\*2 == x;

TRACE of q1\_2012.PrefixSumRec.div2(int)

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:73: requires x > 0;

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:74: requires even(x);

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:82: return x / 2;

VALUE: x === 1

VALUE: 2 === 2

VALUE: x / 2 === 0

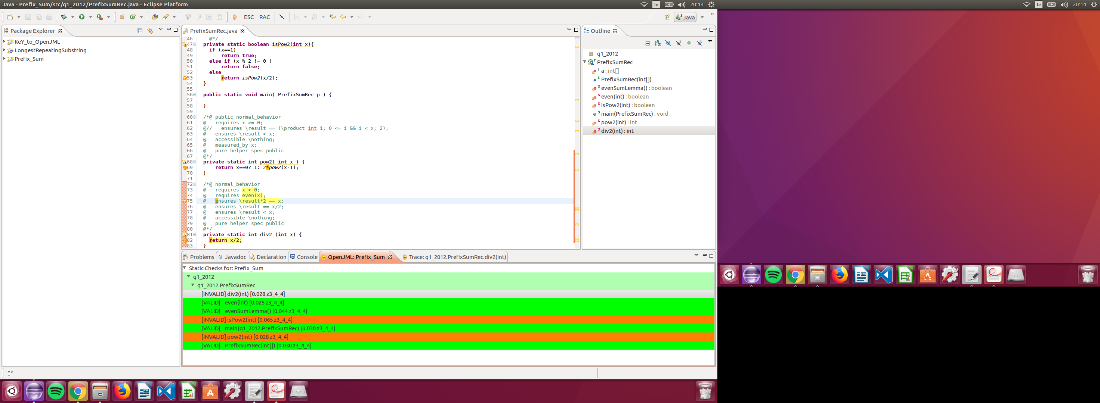
/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:82: ArithmeticOperationRange assertion: !(x == -2147483648 && 2 == -1)

VALUE: !(x\_2233\_0\_\_\_1 == -2147483648 && 2 == -1) === true

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:75: ensures \result \* 2 == x;

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:82: Invalid assertion (Postcondition)

: /home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:75: Associated location



20:21 – div2 method fixed, isPow2 method removed, pow2 error

* + pow2 method: return x==0? 1: 2\*pow2(x-1);
    - ArithmeticOperationRange exception

TRACE of q1\_2012.PrefixSumRec.pow2(int)

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:39: requires x >= 0;

VALUE: x === 1237

VALUE: 0 === 0

VALUE: x >= 0 === true

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:47: return x == 0 ? 1 : 2 \* pow2(x - 1);

VALUE: x === 1237

VALUE: 0 === 0

VALUE: x == 0 === false

VALUE: 2 === 2

VALUE: x === 1237

VALUE: 1 === 1

VALUE: x - 1 === 1236

VALUE: 2 \* pow2(x - 1) === ( - 2147483648 )

VALUE: x == 0 ? 1 : 2 \* pow2(x - 1) === 0

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:47: ArithmeticOperationRange assertion: !(0 < x && 1 < 0) || x <= 2147483647 + 1

VALUE: !(0 < x\_1225\_0\_\_\_1 && 1 < 0) || x\_1225\_0\_\_\_1 <= 2147483647 + 1 === true

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:47: ArithmeticOperationRange assertion: !(x < 0 && 0 < 1) || -2147483648 + 1 <= x

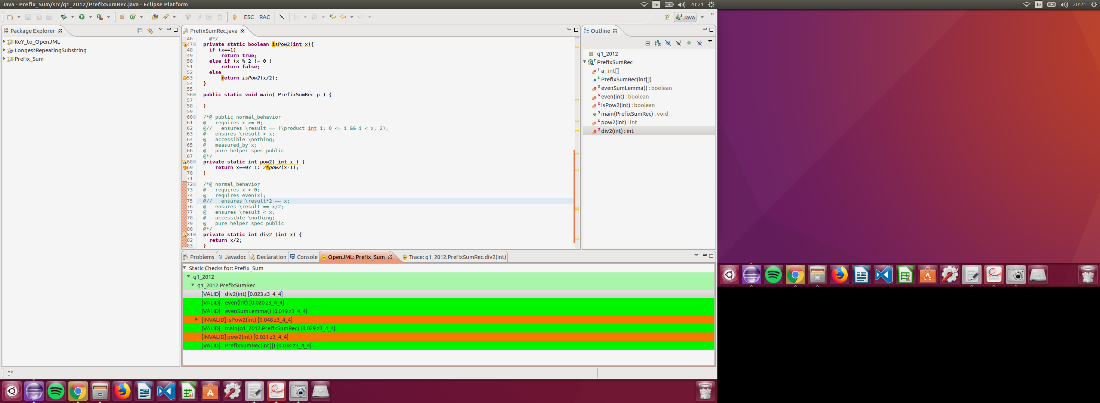
VALUE: !(x\_1225\_0\_\_\_1 < 0 && 0 < 1) || -2147483648 + 1 <= x\_1225\_0\_\_\_1 === true

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:39: Precondition assertion: \_$CPRE\_\_6

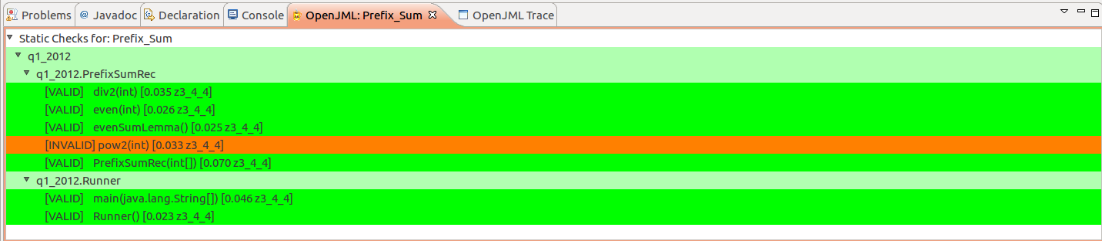
/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:47: ArithmeticOperationRange assertion: -2147483648 <= 2 \* \_JML\_\_tmp71 && 2 \* \_JML\_\_tmp71 <= 2147483647

VALUE: -2147483648 <= 2 \* \_JML\_\_tmp71 && 2 \* \_JML\_\_tmp71 <= 2147483647 === false

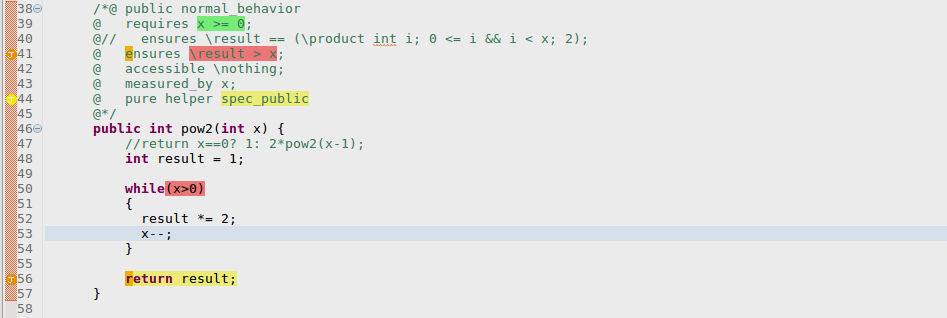
/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:47: Invalid assertion (ArithmeticOperationRange)n



NOTE: isPow2 method is still in the ESC checks from previous run, despite being removed from the program. Need to restart eclipse to remove it from the cache.



21:29 – Changed pow2 method to iterative program but verification still fails. Error in prover?



TRACE of q1\_2012.PrefixSumRec.pow2(int)

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:39: requires x >= 0;

VALUE: x === 0

VALUE: 0 === 0

VALUE: x >= 0 === true

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:48: int result = 1

VALUE: 1 === 1

VALUE: result === 1

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:50: Loop test

VALUE: x === 0

VALUE: 0 === 0

VALUE: x > 0 === false

VALUE: (x > 0) === false

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:56: return result;

VALUE: result === 0

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:41: ensures \result > x;

VALUE: \result === 0

VALUE: x === 0

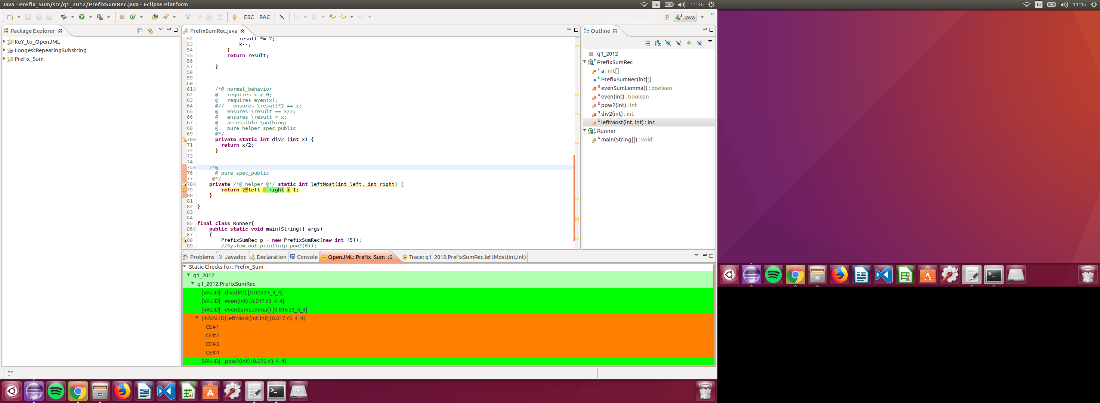
VALUE: \result > x === false

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:56: Invalid assertion (Postcondition)

: /home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:41: Associated location

11:44 – leftMost method

* Arithmetic over and underflow
* Set upper and lower limits to remove error



TRACE of q1\_2012.PrefixSumRec.leftMost(int,int)

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:79: return 2 \* left - right + 1;

VALUE: 2 === 2

VALUE: 2 \* left === 2472

VALUE: right === ( - 2147481175 )

VALUE: 2 \* left - right === 2147483647

VALUE: 1 === 1

VALUE: 2 \* left - right + 1 === ( - 2147483648 )

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:79: ArithmeticOperationRange assertion: -2147483648 <= 2 \* left && 2 \* left <= 2147483647

VALUE: -2147483648 <= 2 \* left\_1773\_0\_\_\_1 && 2 \* left\_1773\_0\_\_\_1 <= 2147483647 === true

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:79: ArithmeticOperationRange assertion: !(0 < \_JML\_\_tmp105 && right < 0) || \_JML\_\_tmp105 <= 2147483647 + right

VALUE: !(0 < \_JML\_\_tmp105 && right\_1783\_0\_\_\_2 < 0) || \_JML\_\_tmp105 <= 2147483647 + right\_1783\_0\_\_\_2 === true

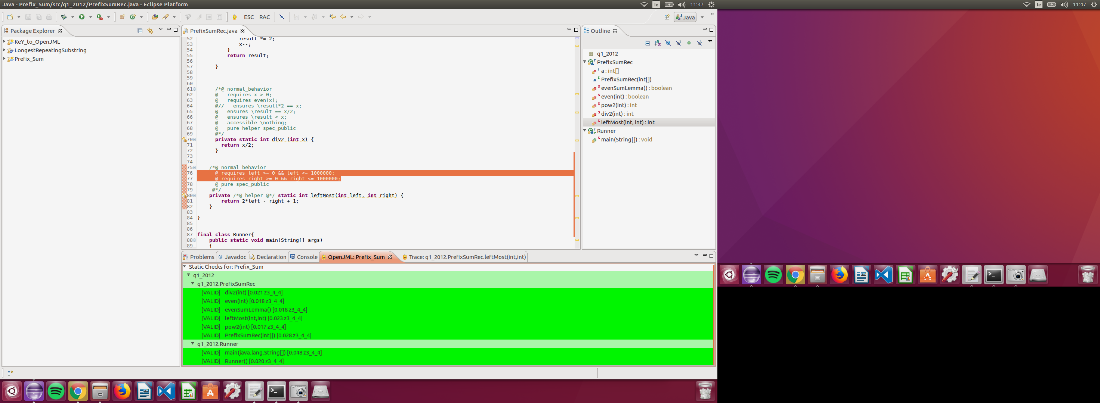
/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:79: ArithmeticOperationRange assertion: !(\_JML\_\_tmp105 < 0 && 0 < right) || -2147483648 + right <= \_JML\_\_tmp105

VALUE: !(\_JML\_\_tmp105 < 0 && 0 < right\_1783\_0\_\_\_2) || -2147483648 + right\_1783\_0\_\_\_2 <= \_JML\_\_tmp105 === true

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:79: ArithmeticOperationRange assertion: !(0 < \_JML\_\_tmp108 && 0 < 1) || \_JML\_\_tmp108 <= 2147483647 - 1

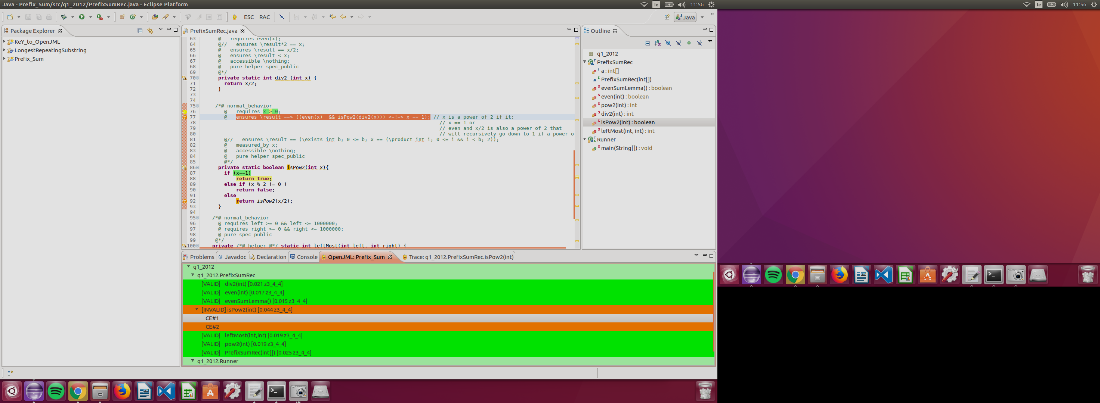
VALUE: !(0 < \_JML\_\_tmp108 && 0 < 1) || \_JML\_\_tmp108 <= 2147483647 - 1 === false

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:79: Invalid assertion (ArithmeticOperationRange)



11:54 – isPow2 method added in again

* Prover cannot establish an assertion
  + LIne:@ ensures \result ==> ((even(x) && isPow2(div2(x))) <=!=> x == 1);
  + Have to remove this line to get past error



TRACE of q1\_2012.PrefixSumRec.isPow2(int)

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:76: requires x > 0;

VALUE: x === 1

VALUE: 0 === 0

VALUE: x > 0 === true

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:87: if (x == 1) ...

VALUE: x === 1

VALUE: 1 === 1

VALUE: x == 1 === true

VALUE: (x == 1) === true

Condition = true

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:88: return true;

VALUE: true === true

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:63: UndefinedCalledMethodPrecondition assertion: \_$CPRE\_\_15

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:76: UndefinedCalledMethodPrecondition assertion: \_$CPRE\_\_18

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:77: Invalid assertion (UndefinedCalledMethodPrecondition)

: /home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:86: Associated location

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

TRACE of q1\_2012.PrefixSumRec.isPow2(int)

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:76: requires x > 0;

VALUE: x === 7188

VALUE: 0 === 0

VALUE: x > 0 === true

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:87: if (x == 1) ...

VALUE: x === 7188

VALUE: 1 === 1

VALUE: x == 1 === false

VALUE: (x == 1) === false

Condition = false

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:89: if (x % 2 != 0) ...

VALUE: x === 7188

VALUE: 2 === 2

VALUE: x % 2 === 0

VALUE: 0 === 0

VALUE: x % 2 != 0 === false

VALUE: (x % 2 != 0) === false

Condition = false

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:92: return isPow2(x / 2);

VALUE: x === 7188

VALUE: 2 === 2

VALUE: x / 2 === 3594

VALUE: isPow2(x / 2) === true

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:92: ArithmeticOperationRange assertion: !(x == -2147483648 && 2 == -1)

VALUE: !(x\_2240\_0\_\_\_1 == -2147483648 && 2 == -1) === true

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:76: Precondition assertion: \_$CPRE\_\_9

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:63: UndefinedCalledMethodPrecondition assertion: \_$CPRE\_\_11

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:77: ensures \result ==> ((even(x) && isPow2(div2(x))) <=!=> x == 1);

VALUE: \result === true

VALUE: x === 3594

VALUE: even(x) === true

VALUE: x === 3594

VALUE: div2(x) === 1797

VALUE: isPow2(div2(x)) === true

VALUE: even(x) && isPow2(div2(x)) === true

VALUE: (even(x) && isPow2(div2(x))) === true

VALUE: x === 3594

VALUE: 1 === 1

VALUE: x == 1 === false

VALUE: (even(x) && isPow2(div2(x))) <=!=> x == 1 === true

VALUE: ((even(x) && isPow2(div2(x))) <=!=> x == 1) === true

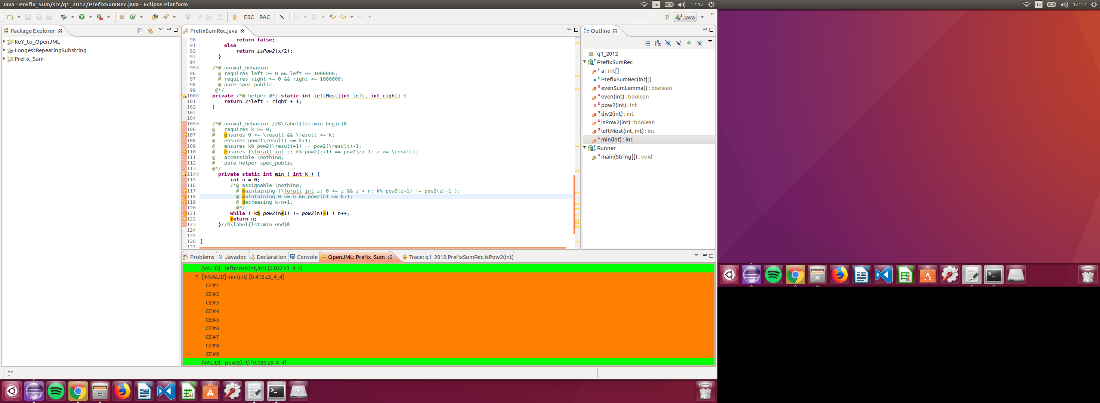
VALUE: \result ==> ((even(x) && isPow2(div2(x))) <=!=> x == 1) === true

/home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:92: Invalid assertion (Postcondition)

: /home/eo37/workspace/Prefix\_Sum/src/q1\_2012/PrefixSumRec.java:77: Associated location

12:09 – min method added in

* Multiple assertion errors
* Multiple arithmetic errors (underflow, overflow, divide by zero)

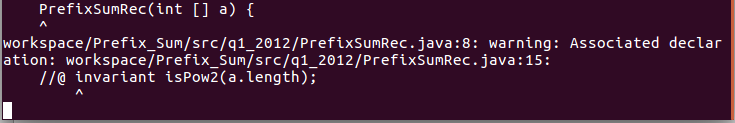
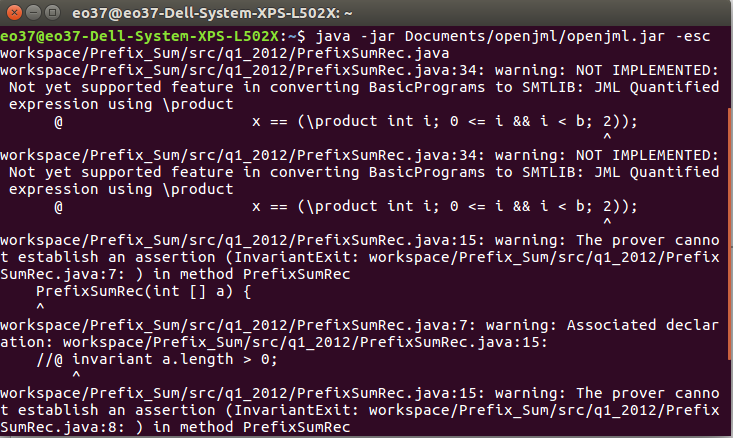


12:52 - min method verification still running, infinite loop?

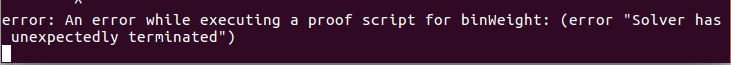
* Line: @ maintaining 0 <= n && pow2(n) <= k+1; has to be removed

### Terminal

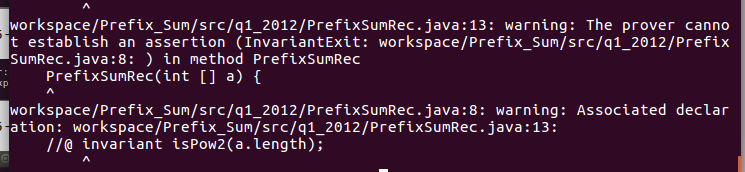
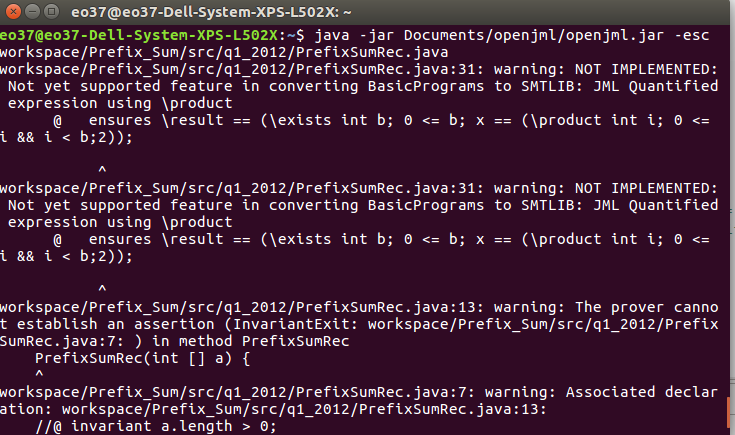
14:42



14:46 – Solver terminates unexpectedly working on binWeight method



15:46 – Second terminal attempt



15:48 Third Terminal attempt

* Error on invariant a.length>0
  + Changed the assertion from an 'invariant' to 'ensures' and applied to constructor
  + Variable name 'a' renamed to 'array' to stop naming difficulties with prover
  + array.length>0 added to constructor
  + array!=null assertion added to constructor
  + @ ensures a.length>0 && a!= null added to constructor
* Error on invariant isPow2(a.length) removed completely, can't be enforced so is not needed.
* binWeight method removed, not used by KeY solution so is not needed.
* \product assertion removed, not sure if needed?