Formally Frama-C

Musa Al-hassy

McMaster University
alhassm@mcmaster.ca

January 13, 2017

A nice number for a taxicab

What is the smallest number representable in two ways as a sum of two cubes?

Exploring The Frama-C World

Recall that Frama-C is an open source Framework for Modular Analysis of C programs whose first public release was in 2008.

- Website `http://frama-c.com`
- Mailing list `http://lists.gforge.inria.fr/cgi-bin/mailman/listinfo/frama-c-discuss`
- Wiki `https://bts.frama-c.com/dokuwiki/doku.php?id=mantis:frama-c:start`
- Blog `http://blog.frama-c.com/`

Main Frama-C plugins

- Value Analysis: static verification using “abstract interpretation”
- WP: static verification using “weakest precondition” —Jessis is a similar plug-in.
- RTE: run time error analysis
- InOut: cmputation of outputs from inputs
- Aorai: temporal verification
- PathCrawler: test generation
- SpareCode: removes spare code

ANSI/ISO-C Specfication Language

ACSL is the specification language for Frama-C and its annotations appear as special comments: `/*@ <reasoning here> */`

E-ACSL: "Executable" ACSL —why can this be dangerous?

\result is not a variable!

We can only assert properties of variables and how Frama-C rewrites some code to make this explicit.

```
/*@
@ assigns \nothing;
@ ensures \result == 42;
*/
int life()
{
  return 42;
}
```

Arguments are copied onto the stack

Erroneous Specification

```
#include<stdio.h>

/*@
@ assigns a;
@ ensures a == 1729;
*/
void setA(int a){ a = 1729; }

int main()
{
  int a = 1 + 12;
  printf("Pre: a = %d\n", a);
  setA(a);
  printf("Post: a = %d\n", a);
}
```

\old vs \at(-,-)

imperative programming is temporal

The built-in Frama-C construct \at refers to the value of a variable at a given point in time.

Logic Labels

- `Here` the position to where the assertion appears.
- `Old` the pre-state of a method and may only appear in its specification.
- `Post` the post-state of a method and may only appear in its specification.
- `LoopCurrent` refers to the state at the beginning of the current step of the loop; it may only appear within a loop body.

Syntactic sugar: `\old(var) == \at(var, Old)`

Printing and Frama-C

Do not use any printf statements in code you want Frama-C to analyse! Print in your driver program, eg `main`.

If you find any more surprising issues, please let me know!

Hoare Calculus in ACSL/Frama-C

Within Frama-C, the WP plug-in enables deductive verification of C programs that have been annotated with ACSL. The WP plug-in uses Hoare-style weakest precondition computations to formally prove ACSL properties of C code. Verification conditions are generated and submitted to external automatic theorem provers or interactive proof assistants. – ACSL By Example §2

$\{Q\}\, S\, \{R\}$	$\equiv$	`//@ assert Q;`
		`S;`
		`//@ assert R;`

Note the new semicolon in the latter version! GCL uses semicolons as separators for catenation, sequencing, of code —as is done with its usage in colloquial English!— rather than a terminator!

Recalling a problem discussed in class

Sorting three variables

$\{true\}$
do
$[] \; y < x \rightarrow x, y := y, x$
$[] \; z < y \rightarrow y, z := z, y$
od
$\{x \leq y \leq z\}$

```
#include "alhassy_gcl.h" // for GCL macros

/*@
@ requires \valid(x) && \valid(y) && \valid(z);
@ assigns *x , *y, *z;
@ ensures *x <= *y <= *z;
*/
void sorting(int* x, int* y, int* z)
{
   //@ loop assigns *x, *y, *z;
    DO
    guard *y < *x has swap(*x, *y) //@ assert *x < *y ;
    guard *z < *y has swap(*y, *z) //@ assert *y < *z ;
    OD
}
```

What more can be added to this specfication?

Try it out!

```
int main()
{
  int x = 9, y = 8, z = 7;
  printf("x,y,z = %d,%d,%d\n", x, y, z);
  sorting(&x, &y, &z);
  printf("x,y,z = %d,%d,%d\n", x, y, z);
}
```

References

ACSL By Example

`http://www.cs.umd.edu/class/spring2016/cmsc838G/frama-c/ACSL-by-Example-12.1.0.pdf`

Highly-recommended! Teaches ACSL accessibly by using examples as the motivator.

ACSL Mini-tutorial

`https://frama-c.com/download/acsl-tutorial.pdf`

"For an in-depth understanding of ACSL, we strongly reccommend users to read the official Frama-C introductory tutorial first." –ACSL By Example

Frama-C reference manual

`https://frama-c.com/download/frama-c-user-manual.pdf`

Explains what is Frama-C and how to get it set up.

ACSL: ANSI/ISO C Specfication Language

`https://frama-c.com/download/acsl.pdf`

This' the ACSL reference document