frama C

Software Analyzers

Where we are, where we go

Allan Blanchard

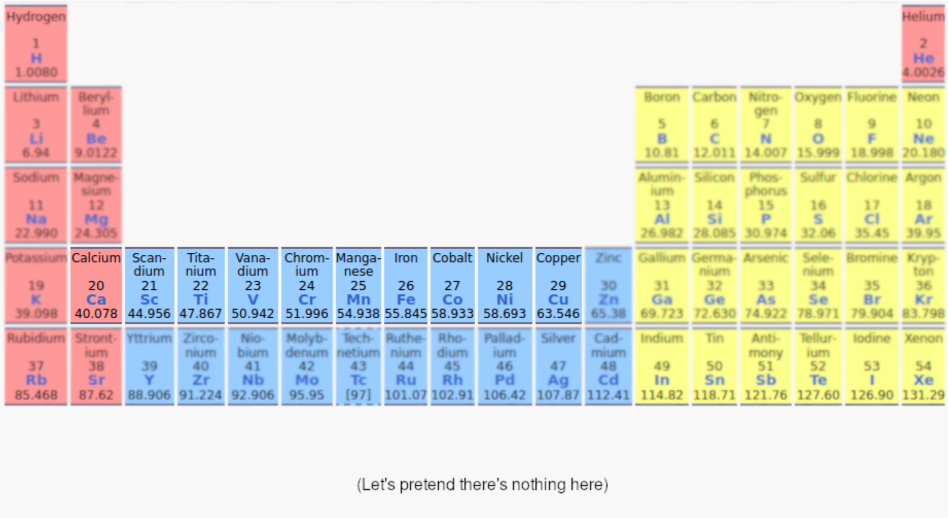
June 13th, 14th @ Frama-C Days

CEA-List, Université Paris-Saclay, Software Safety and Security Lab

Thank you!

- Michele Alberti
- Thibaud Antignac
- Gergö Barany
- Patrick Baudin
- Nicolas Bellec
- Thibaut Benjamin
- Allan Blanchard
- Lionel Blatter
- François Bobot
- Richard Bonichon
- Vincent Botbol
- Quentin Bouillaguet
- David Bühler
- Zakaria Chihani
- Loïc Correnson
- Julien Crétin
- Pascal Cuoq
- Zaynah Dargaye
- Basile Desloges
- Jean-Christophe Filliâtre
- Philippe Herrmann
- Maxime Jacquemin
- Florent Kirchner
- Alexander Kogtenkov
- Rémi Lazarini
- Tristan Le Gall
- Jean-Christophe Léchenet
- Matthieu Lemerre
- Dara Ly
- David Maison
- Claude Marché
- André Maroneze
- Thibault Martin
- Fonenantsoa Maurica
- Melody Méaulle
- Benjamin Monate
- Yannick Moy
- Pierre Nigron
- Anne Pacalet
- Valentin Perrelle
- Guillaume Petiot
- Dario Pinto
- Virgile Prevosto
- Armand Puccetti
- Félix Ridoux
- Virgile Robles
- Jan Rochel
- Muriel Roger
- Julien Signoles
- Nicolas Stouls
- Kostyantyn Vorobyov
- Boris Yakobowski

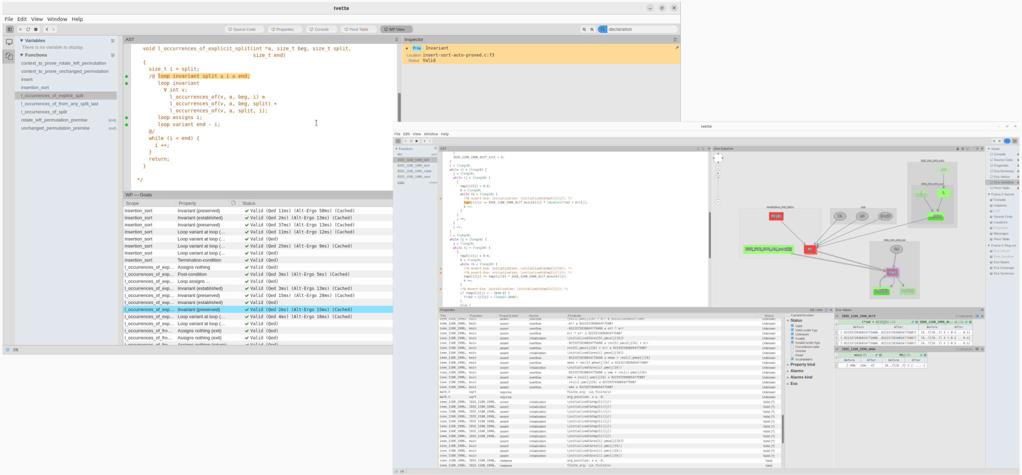
It's been 5 years!



Hydrogen 1 H 1.0080																	Helium 2 He 4.0026
Lithium 3 Li 6.94	Beryllium 4 Be 9.0122											Boron 5 B 10.81	Carbon 6 C 12.011	Nitrogen 7 N 14.007	Oxygen 8 O 15.999	Fluorine 9 F 18.998	Neon 10 Ne 20.180
Sodium 11 Na 22.990	Magnesium 12 Mg 24.305											Aluminium 13 Al 26.982	Silicon 14 Si 28.085	Phosphorus 15 P 30.974	Sulfur 16 S 32.06	Chlorine 17 Cl 35.45	Argon 18 Ar 39.95
Potassium 19 K 39.098	Calcium 20 Ca 40.078	Scandium 21 Sc 44.956	Titanium 22 Ti 47.867	Vanadium 23 V 50.942	Chromium 24 Cr 51.996	Manganese 25 Mn 54.938	Iron 26 Fe 55.845	Cobalt 27 Co 58.933	Nickel 28 Ni 58.693	Copper 29 Cu 63.546	Zinc 30 Zn 65.38	Gallium 31 Ga 69.723	Germanium 32 Ge 72.630	Arsenic 33 As 74.922	Selenium 34 Se 78.971	Bromine 35 Br 79.904	Krypton 36 Kr 83.798
Rubidium 37 Rb 85.468	Strontium 38 Sr 87.62	Yttrium 39 Y 88.906	Zirconium 40 Zr 91.224	Niobium 41 Nb 92.906	Molybdenum 42 Mo 95.95	Technetium 43 Tc [97]	Ruthenium 44 Ru 101.07	Rhodium 45 Rh 102.91	Palladium 46 Pd 106.42	Silver 47 Ag 107.87	Cadmium 48 Cd 112.41	Indium 49 In 114.82	Tin 50 Sn 118.71	Antimony 51 Sb 121.76	Tellurium 52 Te 127.60	Iodine 53 I 126.90	Xenon 54 Xe 131.29

(Let's pretend there's nothing here)

Visualization of analysis results



Ivette
File Edit View Window Help
Source Code
Properties
Console
Pivot Table
WP View
declaration
Variables
There is no variable to display.
Functions
context_to_prove_rotate_left_permutation
context_to_prove_unchanged_permutation
insert
insertion_sort
l_occurrences_of_explicit_split
l_occurrences_of_from_any_split_last
l_occurrences_of_split
rotate_left_permutation_premise (ext)
unchanged_permutation_premise (ext)
AST
Inspector
Prop Invariant
Location insert-sort-auto-proved.c:73
Status Valid
ON

```
void l_occurrences_of_explicit_split(int *a, size_t beg, size_t split,
                                     size_t end)
{
  size_t i = split;
  /*@ loop invariant split ≤ i ≤ end;
      loop invariant
        ∀ int v;
          l_occurrences_of(v, a, beg, i) ≡
          l_occurrences_of(v, a, beg, split) +
          l_occurrences_of(v, a, split, i);
      loop assigns i;
      loop variant end - i;
  @/
  while (i < end) {
    i ++;
  }
  return;
}

*/
```

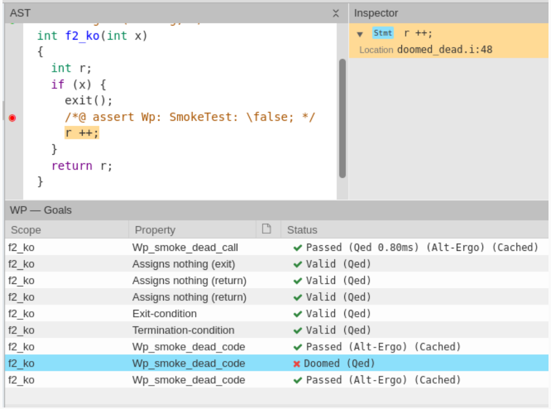
WP — Goals

Scope	Property	Status
insertion_sort	Invariant (preserved)	✓ Valid (Qed 11ms) (Alt-Ergo 58ms) (Cached)
insertion_sort	Invariant (established)	✓ Valid (Qed 2ms) (Alt-Ergo 13ms) (Cached)
insertion_sort	Invariant (preserved)	✓ Valid (Qed 37ms) (Alt-Ergo 13ms) (Cached)
insertion_sort	Loop variant at loop (...	✓ Valid (Qed 11ms) (Alt-Ergo 12ms) (Cached)
insertion_sort	Loop variant at loop (...	✓ Valid (Qed)
insertion_sort	Loop variant at loop (...	✓ Valid (Qed 25ms) (Alt-Ergo 9ms) (Cached)
insertion_sort	Loop variant at loop (...	✓ Valid (Qed)
insertion_sort	Termination-condition	✓ Valid (Qed)
l_occurrences_of_exp...	Assigns nothing	✓ Valid (Qed)
l_occurrences_of_exp...	Post-condition	✓ Valid (Qed 3ms) (Alt-Ergo 5ms) (Cached)
l_occurrences_of_exp...	Loop assigns ...	✓ Valid (Qed)
l_occurrences_of_exp...	Invariant (established)	✓ Valid (Qed 3ms) (Alt-Ergo 13ms) (Cached)
l_occurrences_of_exp...	Invariant (preserved)	✓ Valid (Qed 15ms) (Alt-Ergo 28ms) (Cached)
l_occurrences_of_exp...	Invariant (established)	✓ Valid (Qed)
l_occurrences_of_exp...	Invariant (preserved)	✓ Valid (Qed 2ms) (Alt-Ergo 15ms) (Cached)
l_occurrences_of_exp...	Loop variant at loop (...	✓ Valid (Qed 4ms) (Alt-Ergo 10ms) (Cached)
l_occurrences_of_exp...	Loop variant at loop (...	✓ Valid (Qed)
l_occurrences_of_fro...	Assigns nothing (exit)	✓ Valid (Qed)
l_occurrences_of_fro...	Assigns nothing (exit)	✓ Valid (Qed)
l_occurrences_of_fro...	Assigns nothing (return)	✓ Valid (Qed)

```
  }
  IEEE_1180_1990_dctf_init = 0;
}
i = (long)0;
while (i < (long)8) {
  j = (long)0;
  while (j < (long)8) {
    {
      tmp1[i][j] = 0.0;
      k = (long)0;
      while (k < (long)8) {
        /*@ assert Eva: initialization: \initialized(&tmp1[i][j]); */
        tmp1[i][j] += IEEE_1180_1990_dctf_mcos[k][i] * (double)(*(m1 + k))[j];
        k ++;
      }
    }
    j ++;
  }
  i ++;
}
i = (long)0;
while (i < (long)8) {
  j = (long)0;
  while (j < (long)8) {
    {
      tmp2[i][j] = 0.0;
      k = (long)0;
      while (k < (long)8) {
        /*@ assert Eva: initialization: \initialized(&tmp1[i][k]); */
        /*@ assert Eva: initialization: \initialized(&tmp2[i][j]); */
        tmp2[i][j] += tmp1[i][k] * IEEE_1180_1990_dctf_mcos[k][j];
        k ++;
      }
      /*@ assert Eva: initialization: \initialized(&tmp2[i][j]); */
      if (tmp2[i][j] < - 2048.0) {
        (*(m2 + i))[j] = (long)(-2048);
      }
      else {
```

Feedback on failure

Smoke tests in WP



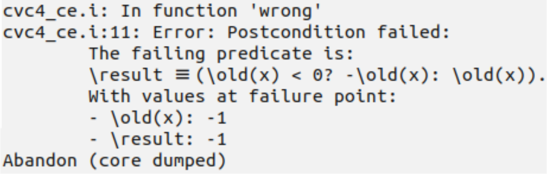
```
int f2_ko(int x)
{
  int r;
  if (x) {
    exit();
    /*@ assert Wp: SmokeTest: \false; */
    r ++;
  }
  return r;
}
```

Stmt r ++;

Location doomed_dead.i:48

Scope	Property		Status
f2_ko	Wp_smoke_dead_call		✓ Passed (Qed 0.80ms) (Alt-Ergo) (Cached)
f2_ko	Assigns nothing (exit)		✓ Valid (Qed)
f2_ko	Assigns nothing (return)		✓ Valid (Qed)
f2_ko	Assigns nothing (return)		✓ Valid (Qed)
f2_ko	Exit-condition		✓ Valid (Qed)
f2_ko	Termination-condition		✓ Valid (Qed)
f2_ko	Wp_smoke_dead_code		✓ Passed (Alt-Ergo) (Cached)
f2_ko	Wp_smoke_dead_code		✗ Doomed (Qed)
f2_ko	Wp_smoke_dead_code		✓ Passed (Alt-Ergo) (Cached)

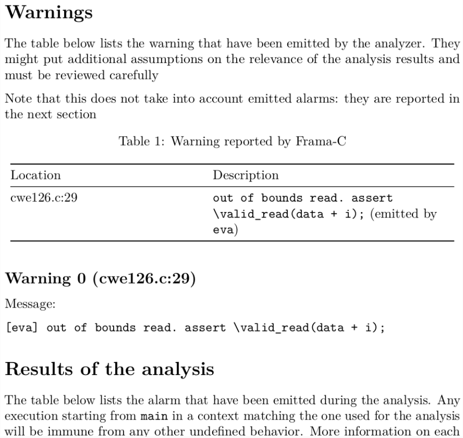
Detailed failure in E-ACSL



```
cvc4_ce.i: In function 'wrong'
cvc4_ce.i:11: Error: Postcondition failed:
        The failing predicate is:
        \result ≡ (\old(x) < 0? -\old(x): \old(x)).
        With values at failure point:
        - \old(x): -1
        - \result: -1
Abandon (core dumped)
```

But also solvers counter examples

Markdown report



Warnings

The table below lists the warning that have been emitted by the analyzer. They might put additional assumptions on the relevance of the analysis results and must be reviewed carefully

Note that this does not take into account emitted alarms: they are reported in the next section

Table 1: Warning reported by Frama-C

Location	Description
cwe126.c:29	`out of bounds read. assert \valid_read(data + i);` (emitted by `eva`)

Warning 0 (cwe126.c:29)

Message:

```
[eva] out of bounds read. assert \valid_read(data + i);
```

Results of the analysis

The table below lists the alarm that have been emitted during the analysis. Any execution starting from `main` in a context matching the one used for the analysis will be immune from any other undefined behavior. More information on each

But also

- JSON output,
- SARIF output.

WP can also provide JSON for proof stats

Analysis

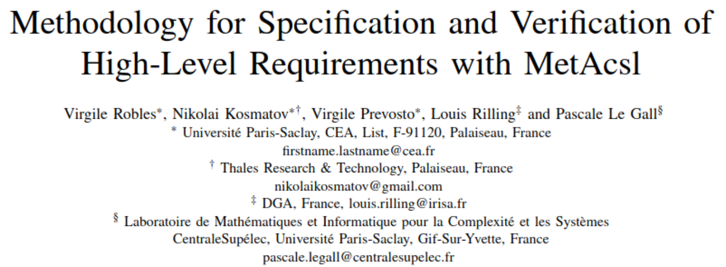
- Many new domains in Eva (octagons, multidim, numerors, taints)
- Proof engineering tools in WP (strategies, tactics, cache, ...)
- Recursive functions, handled in WP, partially handled in Eva

ACSL (and more) Support

ACSL support:

- Ghost typing
- Various improvements in E-ACSL, Eva, WP, ...

High level Specification



Methodology for Specification and Verification of High-Level Requirements with MetAcsl

Virgile Robles[*], Nikolai Kosmatov[*†], Virgile Prevosto[*], Louis Rilling[‡] and Pascale Le Gall[§]
[*] Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
firstname.lastname@cea.fr
[†] Thales Research & Technology, Palaiseau, France
nikolaikosmatov@gmail.com
[‡] DGA, France, louis.rilling@irisa.fr
[§] Laboratoire de Mathématiques et Informatique pour la Complexité et les Systèmes
CentraleSupélec, Université Paris-Saclay, Gif-Sur-Yvette, France
pascale.legall@centralesupelec.fr

Hardware and OS support

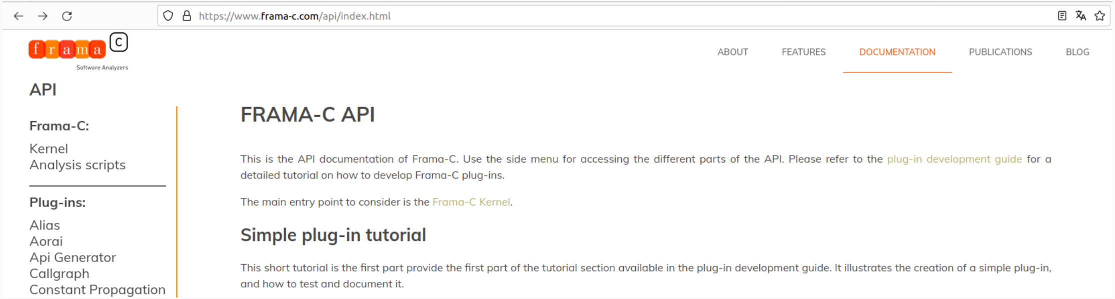
New machdep mechanism:

- Automatic extraction of compiler information
- Customizable YAML file

Runtime E-ACSL

- Can run in multithreaded environment
- Can run on Windows

Just for you, plug-in developers



API

Frama-C:

Kernel
Analysis scripts

Plug-ins:

Alias
Aorai
Api Generator
Callgraph
Constant Propagation

FRAMA-C API

This is the API documentation of Frama-C. Use the side menu for accessing the different parts of the API. Please refer to the plug-in development guide for a detailed tutorial on how to develop Frama-C plug-ins.

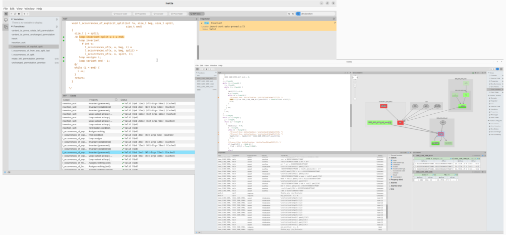
The main entry point to consider is the Frama-C Kernel.

Simple plug-in tutorial

This short tutorial is the first part provide the first part of the tutorial section available in the plug-in development guide. It illustrates the creation of a simple plug-in, and how to test and document it.

- New EVA API
- Alias plug-in

What’s next?



```
void l_occurrences_of_explicit_split(int *a, size_t beg, size_t split,
                                     size_t end)
{
  size_t i = split;
  /*@ loop invariant split ≤ i ≤ end;
      loop invariant
        ∀ int v;
          l_occurrences_of(v, a, beg, i) ≡
          l_occurrences_of(v, a, beg, split) +
          l_occurrences_of(v, a, split, i);
      loop assigns i;
      loop variant end - i;
  */
  while (i < end) {
    i ++;
  }
  return;
}
```

Making Ivette the default Frama-C GUI

- What feature do you miss?
- How should we distribute it?

Enhance specification

- Public contract vs. private contract
- Typestates language and plug-in

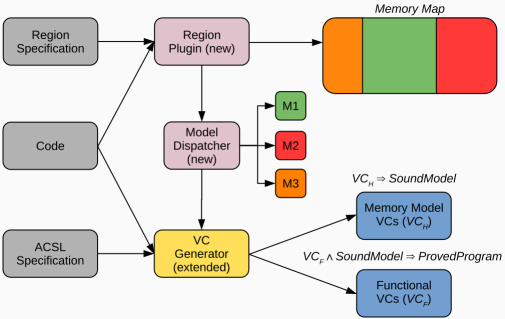
- Concurrent programs analysis is coming :-)

Eva

- Concurrent programs analysis is coming :-)
- Incremental analysis
- Precise analysis for numeric filters
- New, more generic, internal AST

Region memory model

- Better counter examples
- Better Why3 integration
    - Qed + Why3
    - Why3 importer
    - Proof server using Why3find



Memory Map

Region Specification

Region Plugin (new)

Code

Model Dispatcher (new)

M1

M2

M3

ACSL Specification

VC Generator (extended)

$VC_H \Rightarrow SoundModel$

Memory Model VCs (VC_H)

$VC_F \wedge SoundModel \Rightarrow ProvedProgram$

Functional VCs (VC_F)

E-ACSL

- Partial support for axiomatic and inductive definitions
- Labels in predicates and logic functions
- Outline runtime assertion checking
- Performances optimization (static analysis)

E-ACSL

- Partial support for axiomatic and inductive definitions
- Labels in predicates and logic functions
- Outline runtime assertion checking
- Performances optimization (static analysis and optimized code generation)

What if?

We need a new AST

- It needs to be close to the original source code

What if?

We need a new AST

- It needs to be close to the original source code, but
- What if it was generic enough so that it captures more language constructs?

What if?

We need a new AST

- It needs to be close to the original source code, but
- What if it was generic enough so that it captures more language constructs?
- What if we could attach more semantic information to basic constructs?

What if?

We need a new AST

- It needs to be close to the original source code, but
- What if it was generic enough so that it captures more language constructs?
- What if we could attach more semantic information to basic constructs?
- What if it was an ongoing effort?

What if?

We need a new AST

- It needs to be close to the original source code, but
- What if it was generic enough so that it captures more language constructs?
- What if we could attach more semantic information to basic constructs?
- What if it was an ongoing effort?

We target some kind of Frama-All platform

What do you need?

- Deductive proof of programs with dynamic allocation?
- Modular abstract interpretation?
- Runtime assertion checking for concurrent properties?

What do you need?

- Deductive proof of programs with dynamic allocation?
- Modular abstract interpretation?
- Runtime assertion checking for concurrent properties?

Thank you!

It's time to say goodbye

Thank you!

- Jesper Amilon
- Benoît Boyer
- Loïc Correnson
- Tomáš Dacík
- Adel Djoudi
- Marieke Huisman
- Florent Kirchner
- Nikolai Kosmatov
- Julia Lawall
- Matthieu Lemerre
- André Maroneze
- Andrei Paskevich
- Arjtom Plaunov
- Pierre-Yves Piriou
- Samuel Pollard
- Virgile Prevosto
- Franck Sadmi
- Julien Signoles

Thank you!

- Jesper Amilon
- Nanci Naomi Arai
- Wolfgang Ahrendt
- Patrick Baudin
- Thibaut Benjamin
- Nicolas Berthier
- Allan Blanchard
- Lionel Blatter
- François Bobot
- Benoît Boyer
- David Bühler
- Luciana Akemi Burgareli
- Rovedy Aparecida Busquim e Silva
- Cristian Cadar
- David Cok
- Loïc Correnson
- Vincent David
- Mickaël Delahaye
- Adel Djoudi
- Claire Dross
- Zafer Esen
- Jean-Christophe Filliâtre
- Laurent Fuchs
- Arnaud Gotlieb
- Dilian Gurov
- Martin Hána
- Guillaume Hiet
- Marieke Huisman
- Hugo Illous
- Éric Jenn
- Nikolai Kosmatov
- Éric Lavillonnière
- Matthieu Lemerre
- Pascale Le Gall
- Christian Lidström
- Frédéric Loulergue
- Claude Marché
- André Maroneze
- Guillaume Melquiond
- David Mentré
- Raphaël Monat
- David Monniaux
- Laurent Mounier
- Patricia Mouy
- Yannick Moy
- Olivier Nicole
- Jose Maria Parente de Oliveira
- Valentin Perrelle
- Jorge Sousa Pinto
- Pierre-Yves Piriou
- Artjom Plaunov
- Marie-Laure Potet
- Virgile Prevosto
- Xavier Rival
- Philippe Rümmer
- Virgile Robles
- Subash Shankar
- Mihaela Sighireanu
- Julien Signoles
- Laura Titolo
- Franck Védrine
- Virginie Wiels
- Nicky Williams
- Boris Yakobowski

THANK YOU!