Feedback about the experience of Frama-C in SATE VI

André Maroneze and Julien Signoles

Software Reliability & Security Lab

SATE VI Workshop

September 19th, 2019

frama C
Software Analyzers

What is Frama-C?

Framework for analyses of source code written in ISO 99 C

[Kirchner & al in J. of Formal Aspects of Computing 2015]

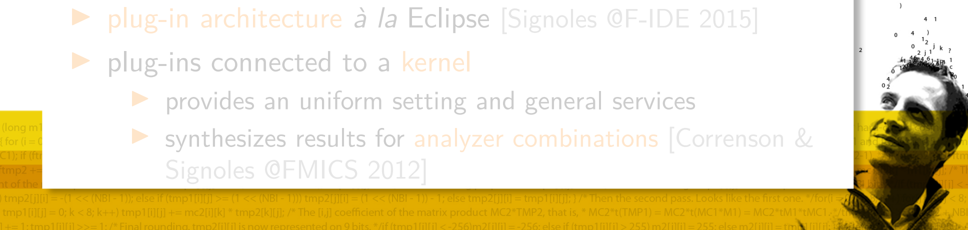
- developed by CEA LIST since 2005
- last open-source release aka 19-Potassium in June 2019

http://frama-c.com

- targets both academic and industrial usages

Several tools inside a single platform

- plug-in architecture *à la* Eclipse [Signoles @F-IDE 2015]
- plug-ins connected to a kernel
  - provides an uniform setting and general services
  - synthesizes results for analyzer combinations [Correnson & Signoles @FMICS 2012]


tmp2[j][i] = -(1 << (NBI - 1)); else if (tmp1[i][j] >= (1 << (NBI - 1))) tmp2[j][i] = (1 << (NBI - 1)) - 1; else tmp2[j][i] = tmp1[i][j]; } /* Then the second pass. Looks like the first one. */for(i =
tmp1[i][j] = 0; k < 8; k++) tmp1[i][j] += mc2[i][k] * tmp2[k][j]; /* The [i,j] coefficient of the matrix product MC2*TMP2, that is, * MC2*t(TMP1) = MC2*t(MC1*M1) = MC2*tM1*tMC1.

What is Frama-C?

Framework for analyses of source code written in ISO 99 C

[Kirchner & al in J. of Formal Aspects of Computing 2015]

- developed by CEA LIST since 2005
- last open-source release aka 19-Potassium in June 2019

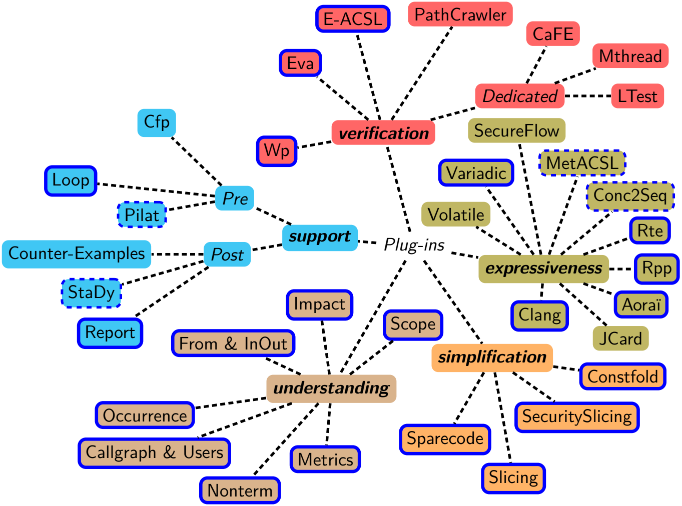
http://frama-c.com

- targets both academic and industrial usages

Several tools inside a single platform

- plug-in architecture *à la* Eclipse [Signoles @F-IDE 2015]
- plug-ins connected to a kernel
  - provides an uniform setting and general services
  - synthesizes results for analyzer combinations [Correnson & Signoles @FMICS 2012]

Frama-C Main Plug-ins



Plug-ins

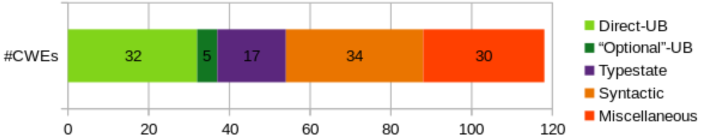
- *verification*
  - E-ACSL
  - PathCrawler
  - Eva
  - Wp
  - *Dedicated*
    - CaFE
    - Mthread
    - LTest
- *support*
  - *Pre*
    - Cfp
    - Loop
    - Pilat
  - *Post*
    - Counter-Examples
    - StaDy
    - Report
- *expressiveness*
  - SecureFlow
  - Variadic
  - MetACSL
  - Conc2Seq
  - Volatile
  - Rte
  - Rpp
  - Aoraï
  - Clang
  - JCard
- *understanding*
  - Impact
  - Scope
  - From & InOut
  - Occurrence
  - Callgraph & Users
  - Metrics
  - Nonterm
- *simplification*
  - Constfold
  - SecuritySlicing
  - Sparecode
  - Slicing

SATE Ockham and Frama-C

- Frama-C focuses on sound, semantic analyses
- Juliet: annotated, extensive, high-quality set of examples
  - Non-regression testing
  - Performance evaluation
- Frama-C in SATE Ockham: Value analysis plug-in
  - Automatic analysis based on abstract interpretation
  - Identifies undefined behaviors (UBs), based on C99/C11
    - No direct CWE identification, but correlated
- Main changes since SATE V
  - Value → Eva (Evolved Value Analysis)
    - More precise and extensible abstract domains
  - Improved handling of several libc functions

Informal classification of Juliet's CWEs

- Informal because CWEs not mathematically exact



- "Optional"-UB: underspecified behaviors
- Typestate: can be found using typestate analyses
  - e.g. input sanitization, access control
- Syntactic: require external (non-ISO C99) input
  - e.g. blacklists, coding conventions
- Miscellaneous: not directly related to UBs
  - e.g. weak PRNG, logic time bombs

Some technical challenges encountered

- Balancing between automation and configurability
  - Typical industrial use case for Eva: large monolithic analysis
    - Dozens of options to customize precision/efficiency
    - For SATE VI Ockham: a single set of options for *all* tests
- Scaling up to 40k+ tests
  - Frama-C initialization time usually negligible (0.*x* seconds)
  - Juliet: 77k C tests, 60% handled by Frama-C (46k)
    - Custom option added to Frama-C, to improve startup

Issues found in Frama-C and by Frama-C

- Issues in Frama-C
  - Documentation: clarifications and reproduction instructions
  - A few edge cases related to string handling
    - Code patterns not seen outside Juliet
    - Fixes applied to Frama-C 18
  - Standard library issues
    - Improvements arriving on Frama-C 20 (Calcium)
- Issues found by Frama-C
  - Accidental CWE: some tests in CWE843 containing CWE562 (out-of-scope use)
  - Unintentional overflow in a test designed to prevent UB

Quick look at SATE 6 Classic track

- Wireshark: library dependencies (glib, epan, etc.) require substantial stubbing effort or integration of several files
- DARPA CGC tests: issues with custom standard library
  - `cgc_libc` requires substantial (and repetitive) renaming and stubbing to use Frama-C's standard library
  - Inclusion of specifications for functions equivalent to `read`/`write`, etc. requires rewriting to enable reuse
    - Each test case has its (incompatible) own version
  - Still, some bugs were found in test cases
    - Started reporting them to TrailOfBits

Conclusion and Perspectives

- SATE Ockham (and Classic track, via DARPA CGC) contributed to Frama-C (test cases, scalability, usability, documentation)
- Frama-C contributed (a bit) to Juliet and SATE
- Reproducible results at: `https://github.com/Frama-C/SATE-VI`
- Perspectives
  - Better precision and CWE coverage
  - SARIF integration
  - Extension to dynamic analysis tools?
    - Experimentation done on SARD-100 with Frama-C/E-ACSL, Google' sanitizers, and RV-Match [Vorobyov, Kosmatov & Signoles @TAP 2018]