Lessons Learned from Verifying Actual C Code with Frama-C

PRiML Workshop

Virgile Prevosto
virgile.prevosto@cea.fr

Université Paris-Saclay, CEA, List

2021-07-12

Outline

Frama-C at a glance

- https://frama-c.com/
- A Framework for modular analysis of C code.
- Initially developed at CEA List and Inria
- Kernel based on CIL (Necula et al. – Berkeley).
- Released under LGPL license (v23.0 Vanadium in June 2021)
- ACSL annotation language.
- Extensible platform
  - Collaboration of analyses over same code
  - Inter plug-in communication through ACSL formulas.
  - Adding specialized plug-ins is easy

Main Frama-C plugins

included in the distribution

available under open-source licence

ANSI/ISO C Specification Language

Presentation

- Based on the notion of contract, like in Eiffel
- Allows users to specify functional properties of their code
- Allows communication between various plugins
- Independent from a particular analysis
- ACSL reference manual at
  https://github.com/acsl-language/acsl

Example of ACSL Specification

```
/*@
requires ptr_val: \valid(a) && \valid(b);
requires ptr_sep: \separated(a,b);
ensures a_val: \at(*a,Pre) == *b;
ensures b_val: \at(*b,Pre) == *a;
*/
void swap(int* a, int* b);
```

Absence of Runtime Errors

- Long-term partnership with EdF and Areva/Framatome
- Fuelled many developments in the Eva plug-in
- [Nucl. Eng. Tech., 2015]

Image ©Daniel Jolivet CC-BY-2.0

Absence of Runtime Errors (2)

- Journey to a RTE-free X.509 parser
- [SSTIC'19]
- Analysis conducted by ANSSI
- Mixing Eva and WP plug-ins
- More complex structures than embedded code

Verification of Functional Properties

- Long-term partnership with Airbus
- Fuelled many developments in the WP plug-in
- Fully integrated in the software toolchain
- Including some internally-developed plug-ins
- [ERTS'20]

Verification of Functional Properties (2)

- Experiments on the Contiki OS
- Part of the H2020 project Vessedia
- Uses WP to prove correctness of the List module
- Ghost code as intermediate between code and spec
- [SAC'19]

Assessing Security Properties

- Experiments on the WooKey bootloader
- Part of Virgile Robles' PhD (MetAcsl plug-in)
- Integrity: bootloader does not write on data banks
- Confidentiality: bootloader only reads what's needed to compute checksums
- [Formalise'21]

Have a Complete, Parseable Code Base

```
#include "somelib/hdr.h"
...
#if SOMECOND==42
extern
 void builtin_f(
  int x, int y);
#endif
...
```

- How to determine complete list of dependencies?
- Which configuration to use for pre-processing?
- Will there be some vendor-dependent functions?

Implementation-Specific Behaviors

```
#pragma pack(push, 4)
struct S_aligned_4 {
  ...
}
#pragma pack(pop)

...
```

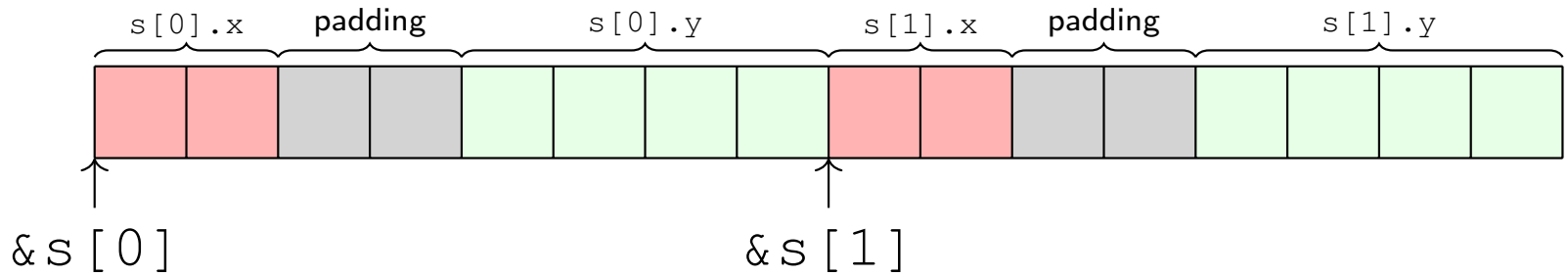
- Many extensions exist beyond ISO C
- Frama-C support added on a case-by-case basis

Intricacies of ISO C Standard

```
unsigned char x[1];
x[0] = x[0];
x[0] += 0;
x[0] *= 0;
if (x[0] != x[0]) {
  /* might not
     be dead
     according to
     standard
   */
}
```

- Standard is sometimes extremely arcane
- With unintended interactions between various sections
- Example: a non-volatile location whose value is allowed to vary
- Spoiler alert: Frama-C is a bit conservative and will warn on line 2 about access to uninitialized value

Casts and Memory Models




```
struct S {
short x;
int y;
} s[2];
```

- Each memory block can be used as **`unsigned char`**`[]`
- Padding bytes have a peculiar status
- Potential aliases can be very complex
- Eva allows for very low-level memory operations
- WP relies on a more abstract model

No Tough Design Choices

- The C standard is always right
  - Just need to understand what it means
- Always possible to restrict a plug-in to a subset of C
  - Unsupported cases are more explicit
  - Facilitates writing examples for adding new feature

Plenty of Potential Use Cases

- Just go to Github!
- Or to specific C code repositories (e.g. Juliet test suite)
- Open-source Case Studies
- Very useful also for finding new exercises when teaching
- Pretty much the only way to have examples of meaningful size

Get Help from the Community

- `language-lawyer` tag on StackOverflow
- Plenty of tooling
  - CIL
  - Clang
  - JCDB
  - LSP
  - ...

Attract Partners

- Common ground with anyone who knows C
- Applied research at the heart of CEA List's mission
- Ensure focus is on concrete problems faced by our partners
- Does not preclude looking at more fundamental work

From Documentation to Formal Specification

- Decoder H2020 project
- Provide a unified platform for storing all relevant documents for a given software project
- Use AI models to extract information from informal documents and/or code
- Help user write formal specifications

Frama-C and Continuous Integration

- LEIA project
- Newly funded as part of French Grand Défi Cyber
- Work on scalability and reuse of analysis results over small code changes

Summary

- A mature framework for specifying and verifying properties of C code
- Address both safety and security analysis
- Many areas for enhancements: new abstract domains, modular Eva, new WP region-based memory models, C++ (or other) front-end, higher-level properties, ...
- Integration in the whole software development cycle
- Importance of concrete use-cases to steer development in interesting directions
- Collaborations welcome!