frama C
Software Analyzers

A Value Analysis for C Programs

Pascal Cuoq et al

CEA List

September 21, 2009

frama C
Software Analyzers

Frama-C: a static analysis framework for C

- A static analyzer is a plug-in

Frama-C: a static analysis framework for C

- A static analyzer is a plug-in
- Frama-C and plug-ins are written in OCaml
- Plug-ins can be distributed independently from Frama-C core
- No recompilation of core to integrate a new plug-in
- Plug-ins provide services to each other

Frama-C: a static analysis framework for C

- A static analyzer is a plug-in
- Frama-C and plug-ins are written in OCaml
- Plug-ins can be distributed independently from Frama-C core
- No recompilation of core to integrate a new plug-in
- Plug-ins provide services to each other

Think “The Gimp” or “Eclipse”

http://frama-c.cea.fr/

frama C
Software Analyzers

for Verification

Initial focus of Frama-C: verification of critical embedded code.

Existing plug-ins are **correct**

for Verification

Initial focus of Frama-C: verification of critical embedded code.

Existing plug-ins are **correct**
(each for its own definition of correct)

Abstract Interpretation, Weakest Precondition, Slicer

for Verification

Initial focus of Frama-C: verification of critical embedded code.

Existing plug-ins are **correct**
(each for its own definition of correct)

Abstract Interpretation, Weakest Precondition, Slicer

Establish safety properties

for Verification

Initial focus of Frama-C: verification of critical embedded code.

Existing plug-ins are **correct**
(each for its own definition of correct)

Abstract Interpretation, Weakest Precondition, Slicer

Establish safety properties
if codebase in the right C subset,. . .

for Verification

Initial focus of Frama-C: verification of critical embedded code.

Existing plug-ins are **correct**
(each for its own definition of correct)

Abstract Interpretation, Weakest Precondition, Slicer

Establish safety properties
if codebase in the right C subset,. . .

Implementing a heuristic method as a Frama-C plug-in?

for Verification

Initial focus of Frama-C: verification of critical embedded code.

Existing plug-ins are **correct**
(each for its own definition of correct)

Abstract Interpretation, Weakest Precondition, Slicer

Establish safety properties
if codebase in the right C subset,. . .

Implementing a heuristic method as a Frama-C plug-in?
Sure, go ahead. . .

frama C
Software Analyzers

Value Analysis plug-in

Abstract Interpretation (comparable to Astrée or PolySpace)
Emphasis on pointers, pointer casts,
handling of statically allocated chained structures

Value Analysis plug-in

Abstract Interpretation (comparable to Astrée or PolySpace)
Emphasis on pointers, pointer casts,
handling of statically allocated chained structures

Results recorded for each statement for use by other plug-ins.
100 000 lines of embedded code — 10GiB of memory (peak)

frama C
Software Analyzers

Clarification Questions

?

frama C
Software Analyzers

Controversial Statement

Attention C compilers authors:
Nobody except you cares about your speed benchmarks.

frama C Software Analyzers

Controversial Statement

Attention C compilers authors:
Nobody except you cares about your speed benchmarks.

Current trend in the C compilation world:
take advantage of any little bit of undefinedness in the standard.

- signed arithmetic overflows not giving 2-complement results
- char* assumed not to alias with an int*
- ...

frama C
Software Analyzers