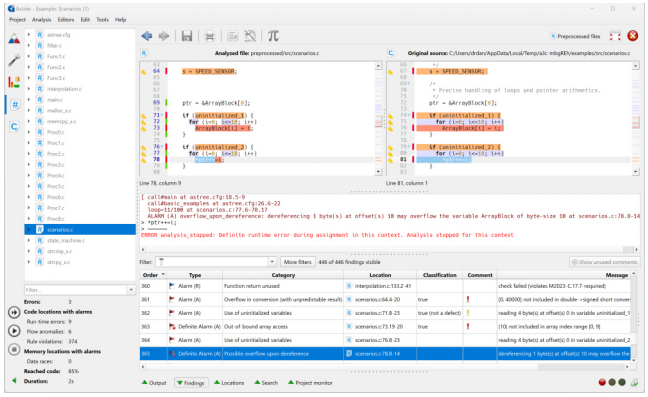
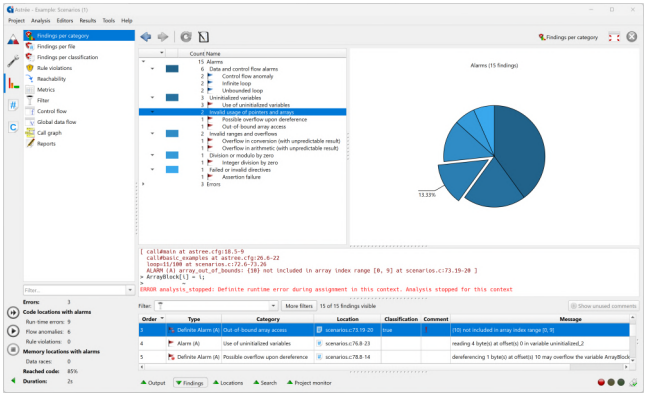
Astrée

Finding all Runtime Errors and Data Races in C/C++ Programs

Astrée is a sound static analyzer designed to **prove the absence of runtime errors and data races** in software programs written in C/C++. Astrée is **parameterizable** and can be **specialized** to the program under analysis – key features to enable **high analysis precision**.





Runtime errors and data races can provoke erroneous program behavior and may even cause the software to crash. They belong to the most dangerous **safety defects** and **cybersecurity vulnerabilities**. Static analysis based on Abstract Interpretation can be used to **prove the absence** of runtime errors and data races. Minimizing the number of false alarms enables an **efficient verification process**.

Examples for Errors detected by Astrée:

- Out-of-bound array accesses
- Erroneous pointer manipulations and dereferencing (NULL, uninitialized, dangling, misaligned, … pointers)
- Divisions by zero and arithmetic overflows
- Read accesses to uninitialized variables
- Pure virtual function calls, invalid `this` pointers
- Memory leaks
- Data races, inconsistent locking, and deadlocks

Additional Features:

- Identify unreachable code
- Detect non-terminating loops
- Find SPECTRE vulnerabilities
- Prove functional properties by static assertions
- User-configurable taint analysis
- Compute inter- and intra-thread control flow graph
- Generate inter- and intra-thread data flow reports
- Software component dependence analysis

Astrée Use Cases:

- **Runtime error analysis** to detect safety and cybersecurity defects at **software component and integration level**
- Report violations of **coding guidelines** to prevent potential safety and security risks
- Demonstrate **freedom of interference** between software components at source code level
- **Signal flow analysis** to prove independence of output signals from input signals
- Contribute to **functional verification** and verify software contracts
- Verify **data and control flow**, enable **data and control coupling** analysis

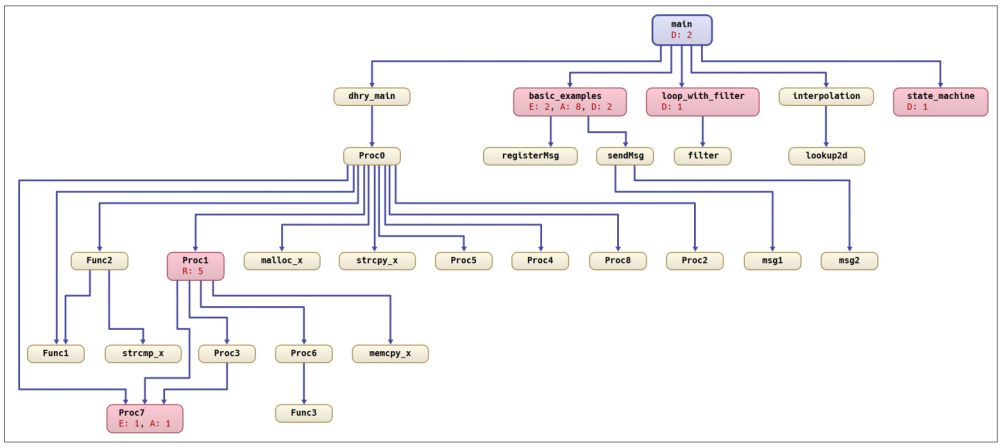
Benefits:

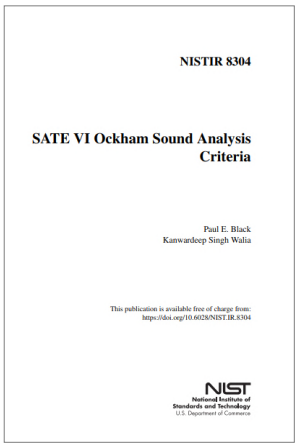
- Satisfy verification objectives of **functional safety**: DO-178C, ISO 26262, IEC 61508…
- Satisfy verification objectives of **cybersecurity**: ISO 21434, ISO/IEC 15408, …
- Demonstrate compliance to **coding guidelines**: MISRA C/C++, CERT C/C++, CWE, …
- Demonstrate **memory safety** at source code level
- **No missed defects**
- **Minimal false positives**
- Full **CI/CD** support
- Automatic **tool qualification**

Astrée is developed and distributed by AbsInt, under license from the CNRS/ENS. It has been successfully used on safety-critical and security-relevant software from various industry sectors, including aerospace, automotive, medical products, and nuclear energy.

AbsInt

AbsInt Angewandte Informatik GmbH Science Park 1 D-66123 Saarbrücken Germany
phone: +49 681 383 60 0 fax: +49 681 383 60 20 info@absint.com www.absint.com





NISTIR 8304

SATE VI Ockham Sound Analysis Criteria

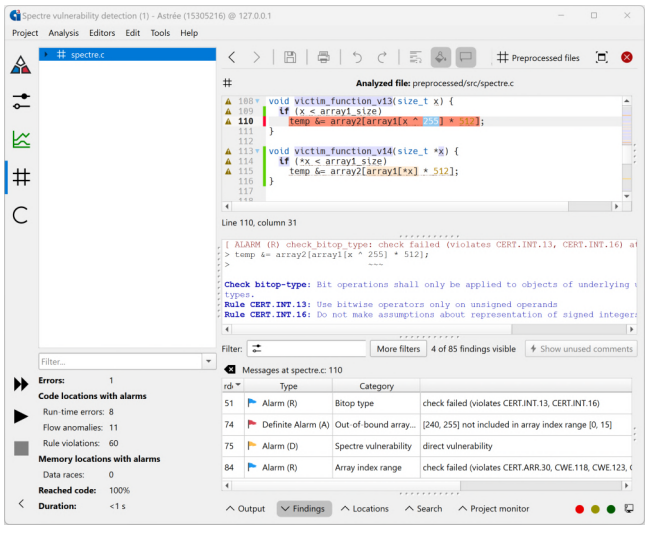
Paul E. Black
Kanwardeep Singh Walia

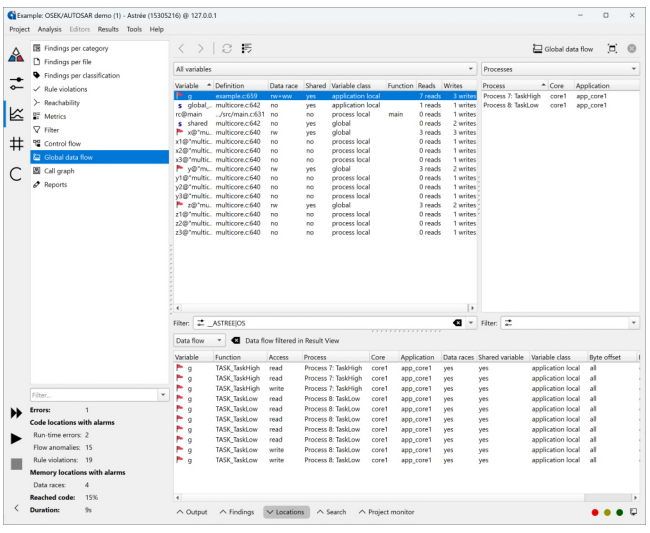
This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8304

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Key Features of Astrée:

- Astrée is **sound**:
  - If the analysis does not detect any runtime errors and data races, their absence has been proven.
  - All possible targets of data and function pointers are taken into account.
  - All possible thread interleavings are considered.
  - Control and data coverage is 100%.
- Astrée is **precise**: Its state-of-the-art analysis engine enables very low false alarm rates.
- **Zero alarm goal**: False alarms can be safely eliminated by tuning the precision to the software under analysis.
- Astrée is **scalable**: Projects with more than 10 million lines of code have successfully been analyzed.
- Astrée features a **sound taint analysis**, capable of demonstrating **freedom of interference.**
- Astrée can be seamlessly integrated in **CI/CD** and **DevOps** environments.
- Astrée is **cloud-ready**: Network connections between Astrée servers and clients are **TLS**-encrypted.
- External user authentication via **OAuth 2.0** / **OIDC** is supported.
- OS-aware analysis of **ARINC 653**, **OSEK**, and **AUTOSAR** projects.
- Automatic setup of AUTOSAR integration analyses from **arxml** files.
- Interactive **visualizations** of call graph, signal flow paths, and C++ class graphs.
- Intuitive **source code navigation** and powerful **interactive result exploration**.
- Supported **coding guidelines** include MISRA C:2004, MISRA C:2012, MISRA C:2023, MISRA C++:2008, MISRA C++:2023, Adaptive AUTOSAR C++ Coding Guidelines, Common Weakness Enumeration CWE, SEI CERT C/C++ Coding Standard, ISO/IEC 17961:2012 C Secure.
- Astrée's Qualification Support Kit enables automatic **tool qualification** up to the highest criticality levels, according to DO-178C, ISO 26262, IEC 61508, and other safety norms.
- **Plugins** for dSPACE TargetLink, Jenkins, Eclipse, µVision are available.
- Astrée satisfies the NIST SATE VI Ockham Sound Analysis Criteria with **market-leading** score.





AbsInt

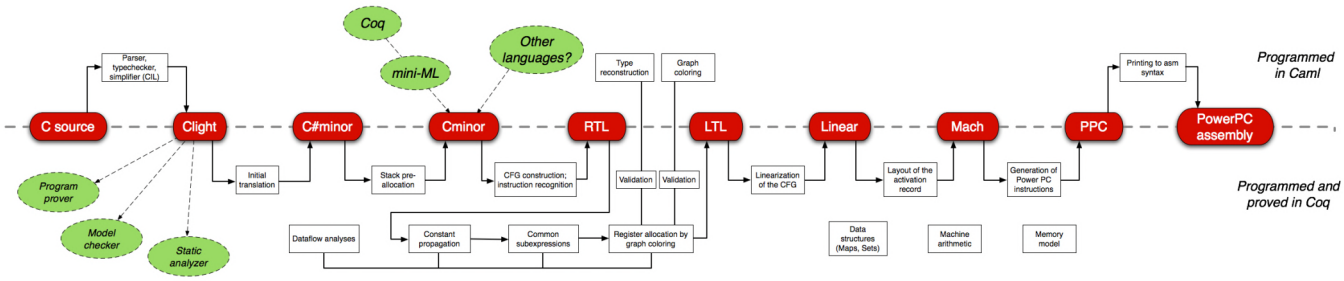
AbsInt Angewandte Informatik GmbH Science Park 1 D-66123 Saarbrücken Germany
phone: +49 681 383 60 0 fax: +49 681 383 60 20 info@absint.com www.absint.com

CompCert

Formally Verified Optimizing C Compiler

ACM Software System Award 2021

CompCert is an **optimizing** C compiler which is **formally verified**, using machine-assisted mathematical proofs, to **guarantee the absence of compiler bugs**. The code it produces is proved to behave exactly as specified by the semantics of the source C program. This level of confidence in the correctness of the compilation process is **unprecedented** and contributes to meeting the **highest software assurance levels**.



CompCert Phases

"The striking thing about our CompCert results is that the middle-end bugs we found in all other compilers are absent. As of early 2011, the under-development version of CompCert is the only compiler we have tested for which Csmith cannot find wrong-code errors. This is not for lack of trying: we have devoted about six CPU-years to the task. The apparent unbreakability of CompCert supports a strong argument that developing compiler optimizations within a proof framework, where safety checks are explicit and machine-checked, has tangible benefits for compiler users."

Study by Regehr, Yang et al. on a development version of CompCert in 2011

In 2021, the **CompCert** development team received the prestigious ACM Software System Award.

Your Benefits:

- Using the **CompCert** C compiler is a natural complement to applying formal verification techniques (static analysis, program proof, model checking) at the source-code level. The correctness proof of **CompCert** guarantees that all safety properties verified on the source code automatically hold for the generated code as well.
- On typical embedded processors, the code generated by **CompCert** usually runs twice as fast as the code generated by GCC without optimizations, and only 20% slower than GCC at optimization level 3.

Availability:

- **CompCert** has been developed at INRIA by architect and lead developer Xavier Leroy with numerous renowned researchers contributing ideas, code, or feedback.
- Source code and documentation of **CompCert**, including the compiler proofs, can be downloaded from the website http://compcert.inria.fr. For research purposes, the usage of CompCert is free of charge.
- In 2014, INRIA and AbsInt entered a license agreement to provide commercial licenses to end users. AbsInt offers commercial licenses, provides industrial-strength support and maintenance, and contributes to the advancement of the tool.

Supported Targets:

- PowerPC (32-bit and 32/64-bit hybrid)
- ARM (ARM & THUMB 32-bit)
- AArch64 (ARM 64-bit)
- IA32 (x86 32-bit)
- AMD64 (x86 64-bit)
- RISC-V (32- and 64-bit)

AbsInt
AbsInt Angewandte Informatik GmbH Science Park 1 D-66123 Saarbrücken Germany
phone: +49 681 383 60 0 fax: +49 681 383 60 20 info@absint.com www.absint.com