Astrée

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

Astrée is a parametric 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.

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

The Challenge:
Runtime errors and data races can provoke erroneous program behavior and may even cause the software to crash. Software testing can be used to detect errors, but not to prove their absence, since usually no complete test coverage is possible. Static analysis based on Abstract Interpretation can be used to prove the absence of runtime errors and data races. A small number of false alarms is important to enable 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)
- Integer and floating-point divisions by zero
- Integer and floating-point arithmetic overflows
- Read accesses to uninitialized variables
- Violations of user-specified assertions
- Pure virtual function calls
- Memory leaks
- Data races between concurrent threads
- Inconsistent locking and deadlocks

Astrée also reports accesses to shared variables, non-terminating loops, and unreachable code. It includes RuleChecker for reporting violations of coding guidelines to prevent potential safety and security risks.

Key Features of Astrée:
- Astrée is sound: If the analysis does not detect any errors, the absence of runtime errors has been proven. Control and data coverage is 100%.
- Astrée is precise: Its state-of-the-art analysis engine enables very low false alarm rates.
- Astrée is scalable: Projects with more than 10 million lines of code have successfully been analyzed.
- False alarms can be safely eliminated by tuning the precision to the software under analysis.
- Astrée can be seamlessly integrated in existing CI and development environments; plugins for Jenkins, Eclipse, µVision and dSPACE TargetLink are available.
- Astrée automatically takes the OS configuration of ARINC653, OSEK, and AUTOSAR projects into account, including the mapping of processes to cores, their resources and their priorities.
- Advanced taint analysis supporting detection of SPECTRE vulnerabilities, and supporting user-defined cybersecurity analyses.
- Astrée supports sound data and control coupling analysis / software component interference analysis.
- Interactive visualizations of call graph and C++ class graph help program review and understanding.
- A Qualification Support Kit is available, providing support for automatic tool qualification up to the highest criticality levels.