Static Analyzer BoF

LLVM Developers' Meeting
October 2015

Static Analyzer Overview

- Supports C / C++ / Objective-C
- Path-sensitive analysis
- Symbolic execution on clang's AST
- Core Engine + Checkers

Core Engine Capabilities

- Simple but fast range-based constraint solver
- Field/offset-sensitive memory model
- Cross-function analysis via "inlining"
- Limited to a single translation unit

Recent Changes

- Widening to improve coverage past loops (Sony)
- New build bot with open source projects
- New checkers
  - `-analyzer-checker=nullability`

Possible Enhancements

- Continue improvements in modeling of loops
- Function factory / modeling of known APIs
- Issue suppression
- Using an SMT solver
- Analysis across translation units
- ...

C++ Support Enhancements

- Behind clang on support for C++11/C++14
  - lambdas
  - brace-initialization
- Teach about STL APIs
- Modeling of temporary objects
- C++ checkers

C++ Temporary Objects

- Destructors must operate on the same region as the constructor
- Pass-by-copy at odds with memory model
- Lifetime extension not modeled correctly

Ecosystem

- http://clang-analyzer.llvm.org
- Open source checker builds
- Documentation:
  - *Building a Checker in 24 Hours* (LLVM Dev 2012)
  - *Checker Developer Manual* on the website
  - `docs/analyzer/` in the clang repository