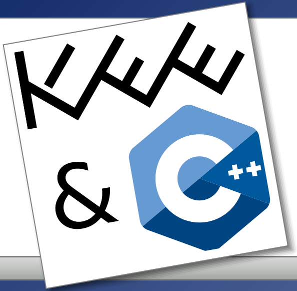

Open-Source C++ Support for KLEE

Felix Rath, Klaus Wehrle

https://comsys.rwth-aachen.de

KLEE'21 Workshop, 2021-06-11

Overview

- Since KLEE 2.0: General support for many C++ programs (libc++ support)

Overview

- Since KLEE 2.0: General support for many C++ programs (libc++ support)
- Since KLEE 2.2: C++ exception handling supported

Overview

- Since KLEE 2.0: General support for many C++ programs (libc++ support)
- Since KLEE 2.2: C++ exception handling supported

- Successfully used it on:

Overview

- **Since KLEE 2.0: General support for many C++ programs (libc++ support)**
- **Since KLEE 2.2: C++ exception handling supported**

- **Successfully used it on:**
  - Z3 - Theorem prover by Microsoft Research

Overview

- **Since KLEE 2.0: General support for many C++ programs (libc++ support)**
- **Since KLEE 2.2: C++ exception handling supported**

- **Successfully used it on:**
  - ▶ Z3 - Theorem prover by Microsoft Research
  - ▶ RE2 - Regular expression library by Google

Overview

- **Since KLEE 2.0: General support for many C++ programs (libc++ support)**
- **Since KLEE 2.2: C++ exception handling supported**

- **Successfully used it on:**
  - ▶ Z3 - Theorem prover by Microsoft Research
  - ▶ RE2 - Regular expression library by Google
  - ▶ jtc - JSON query & manipulation tool

Overview

- **Since KLEE 2.0: General support for many C++ programs (libc++ support)**
- **Since KLEE 2.2: C++ exception handling supported**

- **Successfully used it on:**
  - Z3 - Theorem prover by Microsoft Research
  - RE2 - Regular expression library by Google
  - jtc - JSON query & manipulation tool

- **Publicly available**

Overview

- **Since KLEE 2.0: General support for many C++ programs (libc++ support)**
- **Since KLEE 2.2: C++ exception handling supported**

- **Successfully used it on:**
  - Z3 - Theorem prover by Microsoft Research
  - RE2 - Regular expression library by Google
  - jtc - JSON query & manipulation tool

- **Publicly available**
  - Related work: KLOVER [Li et al., CAV 2011]

C++ in KLEE

- **What is required to support C++ in KLEE?**

C++ in KLEE

- **What is required to support C++ in KLEE?**

- **Frontend compiler**

C++ in KLEE

- **What is required to support C++ in KLEE?**
- **Frontend compiler**
  - ▶ Lowers C++ to LLVM IR

C++ in KLEE

- **What is required to support C++ in KLEE?**

- **Frontend compiler**
  - Lowers C++ to LLVM IR
  - clang++

C++ in KLEE

- **What is required to support C++ in KLEE?**

- **Frontend compiler**
  - ▶ Lowers C++ to LLVM IR
  - ▶ clang++
  - ▶ Fully lowers classes, inheritance, templates, closures, etc.

C++ in KLEE

- **What is required to support C++ in KLEE?**

- **Frontend compiler**
  - Lowers C++ to LLVM IR
  - clang++
  - Fully lowers classes, inheritance, templates, closures, etc.
- **Standard library & runtime**

C++ in KLEE

- **What is required to support C++ in KLEE?**

- **Frontend compiler**
  - Lowers C++ to LLVM IR
  - clang++
  - Fully lowers classes, inheritance, templates, closures, etc.
- **Standard library & runtime**
  - `std::cout, std::vector` etc.

C++ in KLEE

- **What is required to support C++ in KLEE?**

- **Frontend compiler**
  - Lowers C++ to LLVM IR
  - clang++
  - Fully lowers classes, inheritance, templates, closures, etc.
- **Standard library & runtime**
  - `std::cout, std::vector` etc.
  - We use LLVM’s libc++ & libc++abi (threads disabled)

C++ in KLEE

- **What is required to support C++ in KLEE?**

- **Frontend compiler**
  - Lowers C++ to LLVM IR
  - clang++
  - Fully lowers classes, inheritance, templates, closures, etc.
- **Standard library & runtime**
  - `std::cout, std::vector` etc.
  - We use LLVM's libc++ & libc++abi (threads disabled)
  - Requires a libc

C++ in KLEE

- **What is required to support C++ in KLEE?**

- **Frontend compiler**
  - Lowers C++ to LLVM IR
  - clang++
  - Fully lowers classes, inheritance, templates, closures, etc.
- **Standard library & runtime**
  - `std::cout, std::vector` etc.
  - We use LLVM's libc++ & libc++abi (threads disabled)
  - Requires a libc
  - Additional support required: Exception handling

C++ in KLEE

- **What is required to support C++ in KLEE?**

- **Frontend compiler**
  - ▶ Lowers C++ to LLVM IR
  - ▶ clang++
  - ▶ Fully lowers classes, inheritance, templates, closures, etc.
- **Standard library & runtime**
  - ▶ `std::cout, std::vector` etc.
  - ▶ We use LLVM's libc++ & libc++abi (threads disabled)
  - ▶ Requires a libc
  - ▶ Additional support required: Exception handling
- **LLVM support**

C++ in KLEE

- **What is required to support C++ in KLEE?**
- **Frontend compiler**
  - ▶ Lowers C++ to LLVM IR
  - ▶ clang++
  - ▶ Fully lowers classes, inheritance, templates, closures, etc.
- **Standard library & runtime**
  - ▶ `std::cout, std::vector` etc.
  - ▶ We use LLVM's libc++ & libc++abi (threads disabled)
  - ▶ Requires a libc
  - ▶ Additional support required: Exception handling
- **LLVM support**
  - ▶ Previously unimplemented instructions, etc.

Exception Handling

- C++ exception handling based on the Itanium C++ ABI

Exception Handling

- **C++ exception handling based on the Itanium C++ ABI**
  - Unwinding: Platform specific, language independent

Exception Handling

- **C++ exception handling based on the Itanium C++ ABI**
  - Unwinding: Platform specific, language independent
  - C++-rules: Platform independent, language specific

Exception Handling

- **C++ exception handling based on the Itanium C++ ABI**
  - ▶ Unwinding: Platform specific, language independent
  - ▶ C++-rules: Platform independent, language specific
- **KLEE implementation compatible**

Exception Handling

- **C++ exception handling based on the Itanium C++ ABI**
  - Unwinding: Platform specific, language independent
  - C++-rules: Platform independent, language specific
- **KLEE implementation compatible**

- **Unwinding is C++-independent**

Exception Handling

- **C++ exception handling based on the Itanium C++ ABI**
  - ▶ Unwinding: Platform specific, language independent
  - ▶ C++-rules: Platform independent, language specific
- **KLEE implementation compatible**

- **Unwinding is C++-independent**
- **Can be reused by other languages**

Exception Handling

- **C++ exception handling based on the Itanium C++ ABI**
  - ▶ Unwinding: Platform specific, language independent
  - ▶ C++-rules: Platform independent, language specific
- **KLEE implementation compatible**

- **Unwinding is C++-independent**
- **Can be reused by other languages**
  - ▶ New personality function needed

Exception Handling

- **C++ exception handling based on the Itanium C++ ABI**
  - ▶ Unwinding: Platform specific, language independent
  - ▶ C++-rules: Platform independent, language specific
- **KLEE implementation compatible**

- **Unwinding is C++-independent**
- **Can be reused by other languages**
  - ▶ New personality function needed

- **Symbolic exception values supported**

Real-World Results

- **Tried on three-open source C++ projects**

Real-World Results

- Tried on three-open source C++ projects
- 10 runs, 1 hour each, KLEE 2.2

Real-World Results

- Tried on three-open source C++ projects
- 10 runs, 1 hour each, KLEE 2.2

- jtc

Real-World Results

- **Tried on three-open source C++ projects**
- **10 runs, 1 hour each, KLEE 2.2**

- **jtc**
  - ~32.5% instr. cov.

Real-World Results

- **Tried on three-open source C++ projects**
- **10 runs, 1 hour each, KLEE 2.2**

- **jtc**
  - ~32.5% instr. cov.
  - Found three new bugs, and one uncaught exception

Real-World Results

- **Tried on three-open source C++ projects**
- **10 runs, 1 hour each, KLEE 2.2**

- **jtc**
  - ~32.5% instr. cov.
  - Found three new bugs, and one uncaught exception
- **RE2**

Real-World Results

- **Tried on three-open source C++ projects**
- **10 runs, 1 hour each, KLEE 2.2**

- **jtc**
  - ~32.5% instr. cov.
  - Found three new bugs, and one uncaught exception
- **RE2**
  - ~28.6% instr. cov.

Real-World Results

- **Tried on three-open source C++ projects**
- **10 runs, 1 hour each, KLEE 2.2**

- **jtc**
  - ~32.5% instr. cov.
  - Found three new bugs, and one uncaught exception
- **RE2**
  - ~28.6% instr. cov.
  - No bugs found

Real-World Results

- **Tried on three-open source C++ projects**
- **10 runs, 1 hour each, KLEE 2.2**

- **jtc**
  - ~32.5% instr. cov.
  - Found three new bugs, and one uncaught exception
- **RE2**
  - ~28.6% instr. cov.
  - No bugs found
- **Z3**

Real-World Results

- **Tried on three-open source C++ projects**
- **10 runs, 1 hour each, KLEE 2.2**

- **jtc**
  - ~32.5% instr. cov.
  - Found three new bugs, and one uncaught exception
- **RE2**
  - ~28.6% instr. cov.
  - No bugs found
- **Z3**
  - ~0.3% instr. cov.

Real-World Results

- Tried on three-open source C++ projects
- 10 runs, 1 hour each, KLEE 2.2

- jtc
  - ~32.5% instr. cov.
  - Found three new bugs, and one uncaught exception
- RE2
  - ~28.6% instr. cov.
  - No bugs found
- Z3
  - ~0.3% instr. cov.
  - New bug found: `fclose(m_file)` where `m_file` is `NULL`

Conclusion

- Open-source C++ support for KLEE

Conclusion

- Open-source C++ support for KLEE
- Required components

Conclusion

- **Open-source C++ support for KLEE**
- **Required components**
  - Frontend compiler, standard library & compiler runtime, LLVM support

Conclusion

- Open-source C++ support for KLEE
- Required components
  - Frontend compiler, standard library & compiler runtime, LLVM support
- Exception handling in C++

Conclusion

- **Open-source C++ support for KLEE**
- **Required components**
  - Frontend compiler, standard library & compiler runtime, LLVM support
- **Exception handling in C++**
- **Language independent unwinding**

Conclusion

- **Open-source C++ support for KLEE**
- **Required components**
  - Frontend compiler, standard library & compiler runtime, LLVM support
- **Exception handling in C++**
- **Language independent unwinding**
- **Applicable to real-world programs**

Conclusion

- **Open-source C++ support for KLEE**
- **Required components**
  - ▶ Frontend compiler, standard library & compiler runtime, LLVM support
- **Exception handling in C++**
- **Language independent unwinding**
- **Applicable to real-world programs**

- **Thanks to: Lukas Wölfer, Julian Büning, Martin Nowack, Timotej Kapus, Cristian Cadar**

Conclusion

- Open-source C++ support for KLEE
- Required components
  - Frontend compiler, standard library & compiler runtime, LLVM support
- Exception handling in C++
- Language independent unwinding
- Applicable to real-world programs

- Thanks to: Lukas Wölfer, Julian Büning, Martin Nowack, Timotej Kapus, Cristian Cadar

- Try it out on your project!