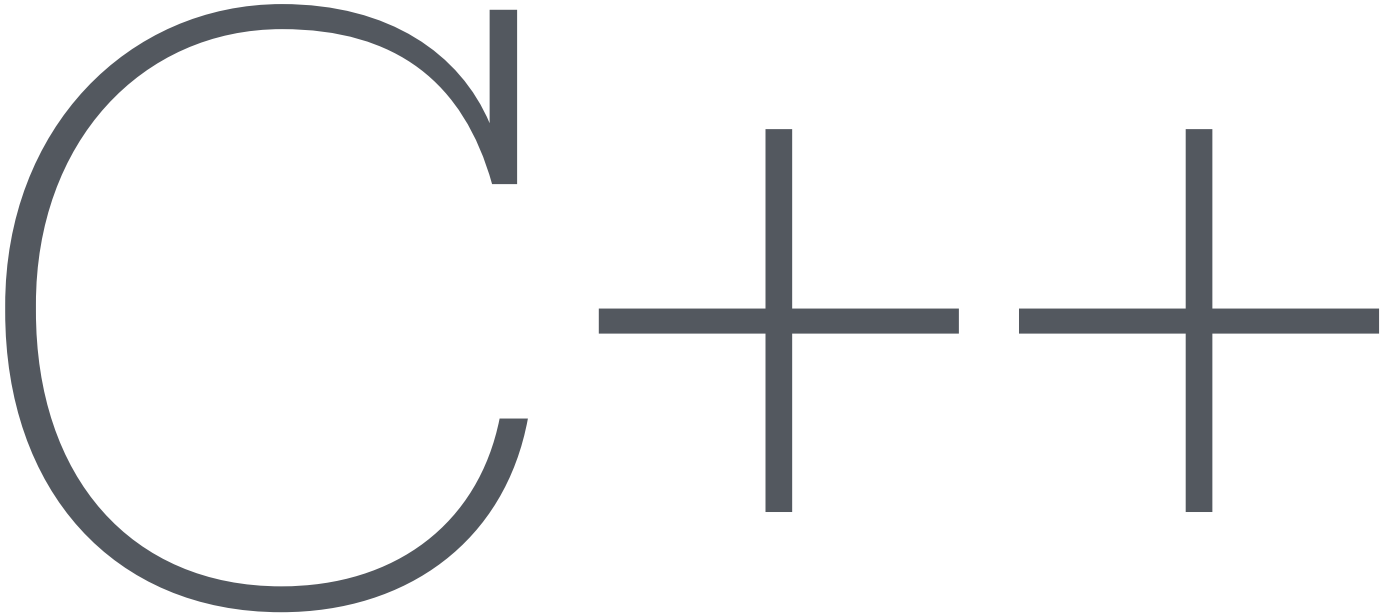
Borrow-Checker


C++

Borrowing

Trouble:

Difficulties


The

65

Authors: danakj@chromium.org, lukasza@chromium.org, palmer@chromium.org
Publication Date: 10th September 2021

Introduction

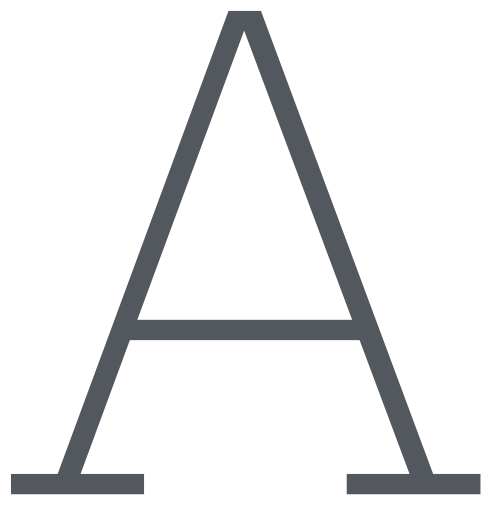
A common question raised when comparing C++ and Rust is whether the Rust borrow checker is really unique to Rust, or if it can be implemented in C++ too. C++ is a very flexible language, so it seems like it should be possible. In this article we'll explore if it is possible to do borrow checking at compile time in C++.

Some background on C++ efforts

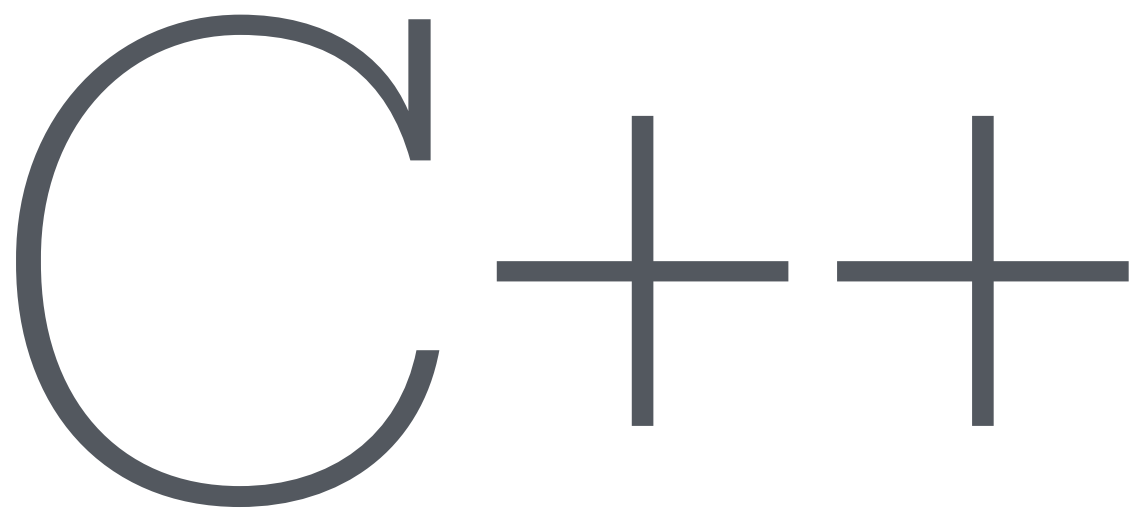
Many folks are working on improving C++, including improving its memory safety. Clang has experimental -Wlifetime warnings to help catch a class of use-after-free bugs. The cases it catches are typically dangling references to temporaries, which makes them a valuable set of warnings to enable when it is available. But the cases it would solve do not seem to intersect with the set of cases MiraclePtr is attempting to protect against, which is an effort to frustrate exploits we've observed in Chrome. MiraclePtr would be used to rewrite and verify pointer dereferences of fields in heap-allocated objects at runtime. This article asks if we could do the same sort of verification at compile time, similar to Rust.

What's a borrow checker?

One tool the Rust compiler uses to ensure the memory safety of a program is its borrow checker. The borrow checker ensures that an object is always in one of 3 states:



Borrow-Checker


C++

66