SMT-based constraint solving in Lean 4

The cvc.lean library: safety and ergonomics

Adrien Champion - *he/him*

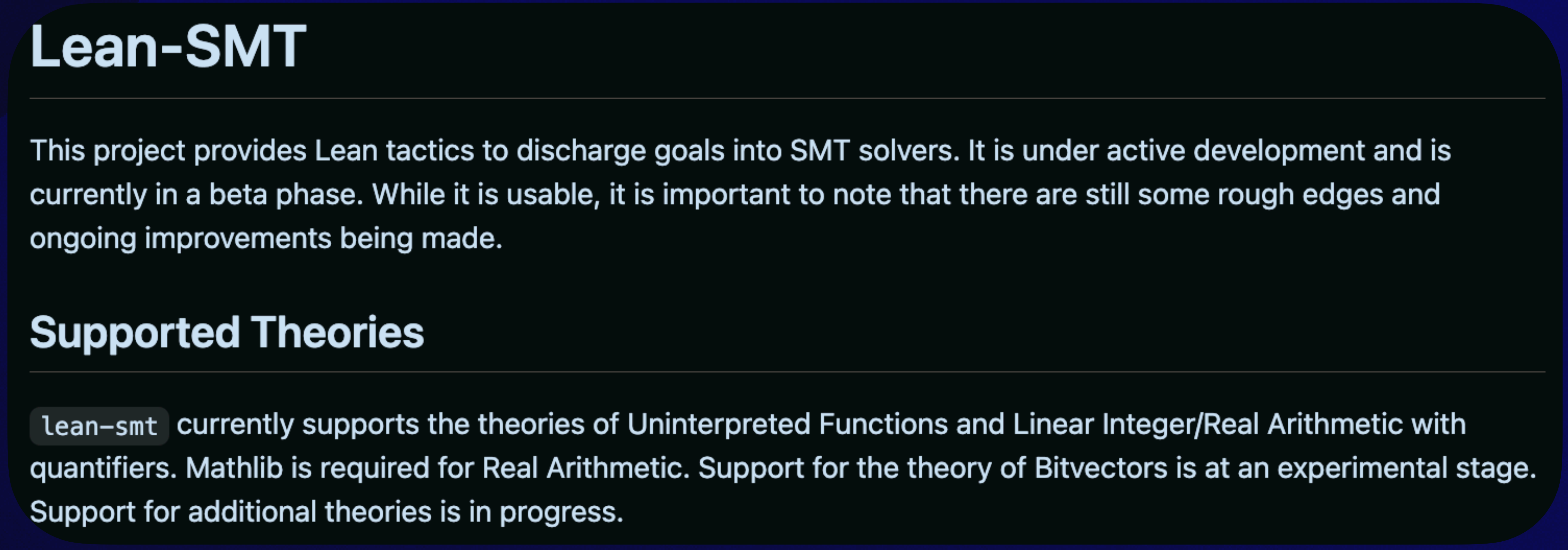
repository
github.com/anzenlang/cvc.lean

information, slides, and relevant links
anzenlang.io

Adrien Champion
adrien.champion@anzenlang.io

cvc.lean: context

- collaboration with *Cesare Tinelli* at the University of Iowa, and the cvc5 team
- Lean 4 library exposing the cvc5 (C++) SMT solver's API --- using C-level FFI
- focus on safety and ergonomics
- public but unstable, not officially released: everything can change (and will improve)
- offshoot of the lean-smt project:



Lean-SMT

This project provides Lean tactics to discharge goals into SMT solvers. It is under active development and is currently in a beta phase. While it is usable, it is important to note that there are still some rough edges and ongoing improvements being made.

Supported Theories

`lean-smt` currently supports the theories of Uninterpreted Functions and Linear Integer/Real Arithmetic with quantifiers. Mathlib is required for Real Arithmetic. Support for the theory of Bitvectors is at an experimental stage. Support for additional theories is in progress.

Let's jump right into it

- **demo time**

- access the (documented) demo file here:

  github.com/anzenlang/cvc.lean/blob/demo_2025_02/CvcTest/Demo/February2025.lean

- or simply go to anzenlang.io/blog

Adrien Champion
adrien.champion@anzenlang.io

Advanced features

- interpolation
- quantifier-elimination
- partial information retrieval in *unknown* mode
- proof / unsat-core retrieval in *unsat* mode
- sat-core in *unsat* mode

Thank you!

Useful links

- information, slides, relevant links for this talk: anzenlang.io
- cvc.lean: github.com/anzenlang/cvc.lean
- lean-cvc5 (very low-level cvc5 FFI): github.com/abdoo8080/lean-cvc5
- lean-smt: github.com/ufmg-smite/lean-smt

Adrien Champion
adrien.champion@anzenlang.io