SMT-based constraint solving in Lean 4

The `cvc.lean` library: safety and ergonomics

Adrien Champion — *he/him*

Adrien Champion
adrien.champion@anzenlang.io

repository
github.com/anzenlang/cvc.lean
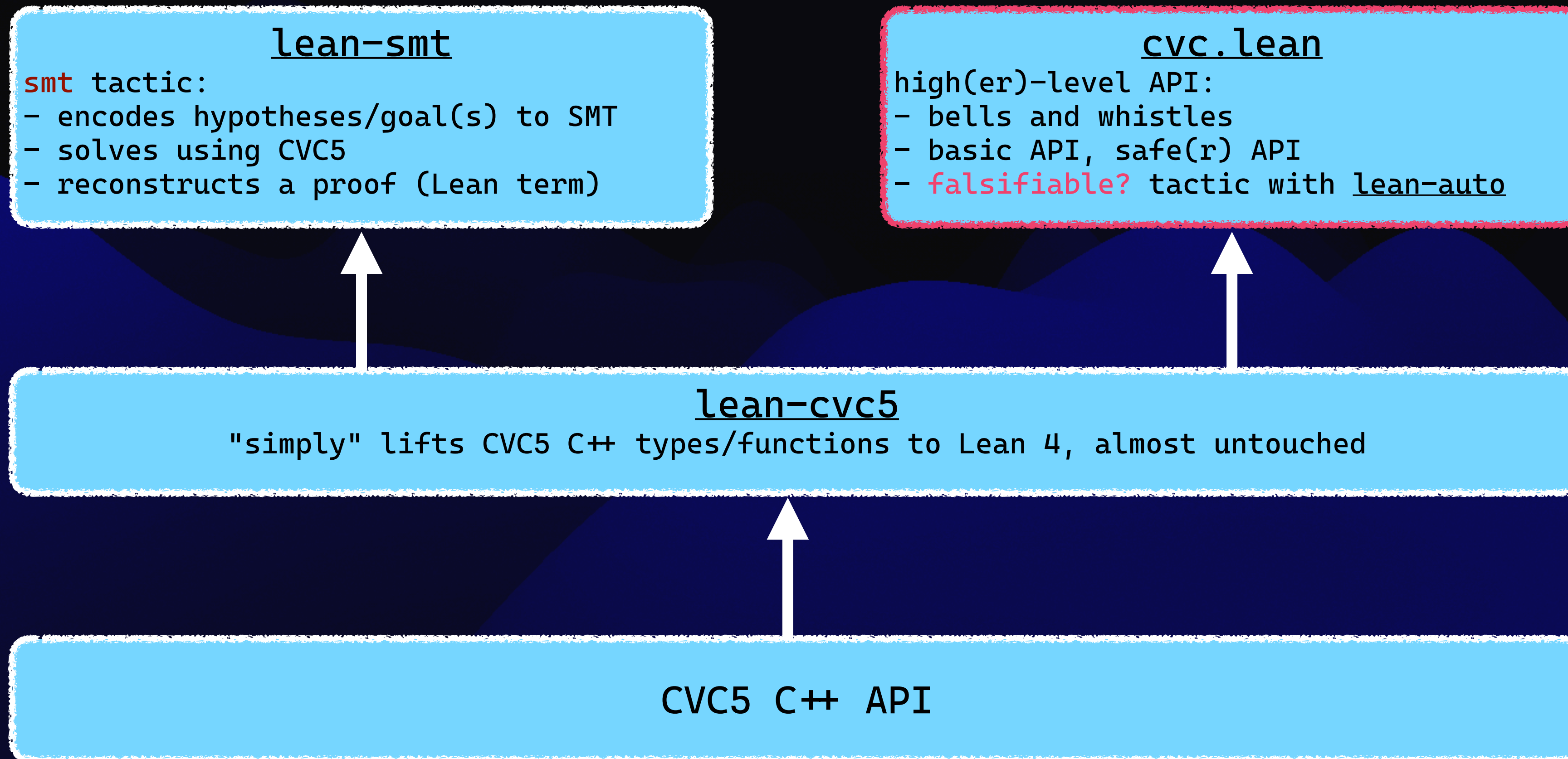
information, slides, and relevant links
anzenlang.io

cvc.lean: context

- collaboration 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

Adrien Champion
adrien.champion@anzenlang.io

cvc5 libraries: architecture

lean-smt

smt tactic:
- encodes hypotheses/goal(s) to SMT
- solves using CVC5
- reconstructs a proof (Lean term)

cvc.lean

high(er)-level API:
- bells and whistles
- basic API, safe(r) API
- falsifiable? tactic with lean-auto

lean-cvc5
"simply" lifts CVC5 C++ types/functions to Lean 4, almost untouched

CVC5 C++ API

Adrien Champion
adrien.champion@anzenlang.io

Let me just show you

- access the (documented) demo file here:

  github.com/anzenlang/cvc.lean/blob/2025_02_demo/CvcTest/Demo/2025February.lean

- or retrieve this link (and the slides) at 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

Adrien Champion
adrien.champion@anzenlang.io

Future work

- expose more of the underlying cvc5 C++ API: String, Array, BitVector...

- stronger constraints on overloaded functions such as add/mul/*etc.*

- push the Safe.SmtM environment further

  - InitM: pre-SmtM, only allows setOption-like commands
    and setLogic returning an SmtM (which would not allow these commands)

  - ask for a proof that produceModels is set when running getValue/getModel?

- more flexible unsafe/safe(r) API-s

  - ergonomic, safe bridges between the two API-s would let users benefit from safety where
    appropriate for their use-case

Adrien Champion
adrien.champion@anzenlang.io

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