ETH zürich

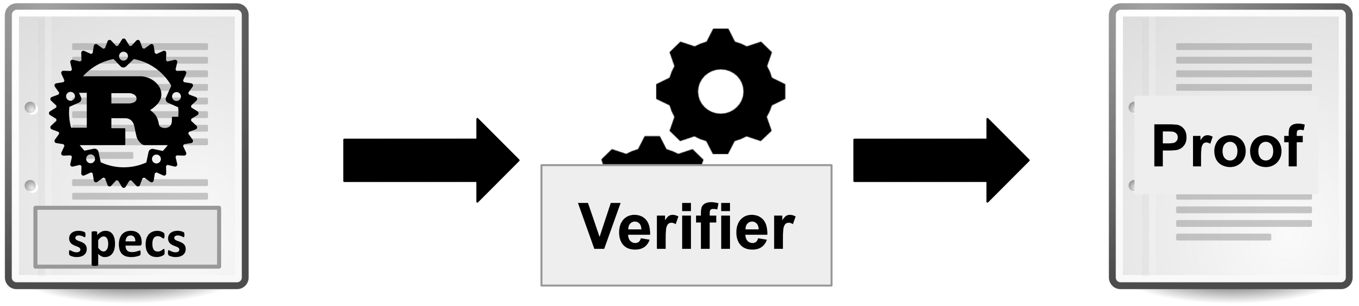
Building User-Friendly Rust Verification Tools

Federico Poli

federico.poli@inf.ethz.ch

@ Rust Verification Workshop 2024

Translation-based automated verifiers



P∗rust→∗i

Viper

Creusot

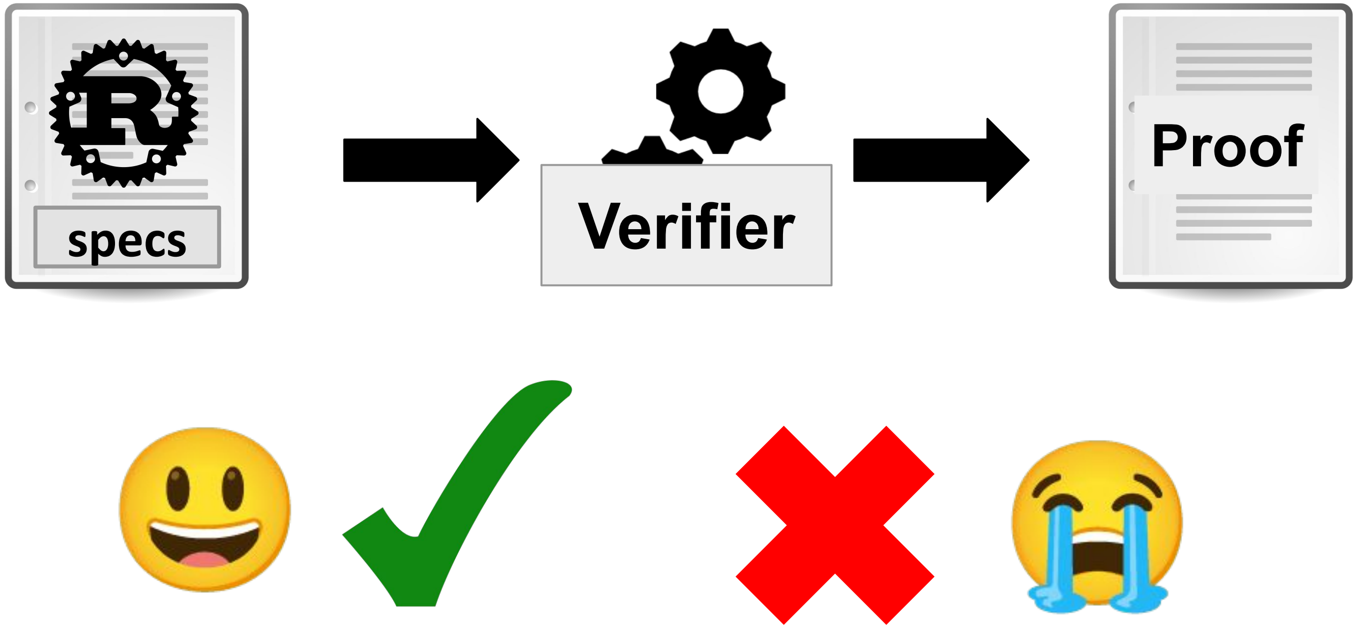
Why3

kani

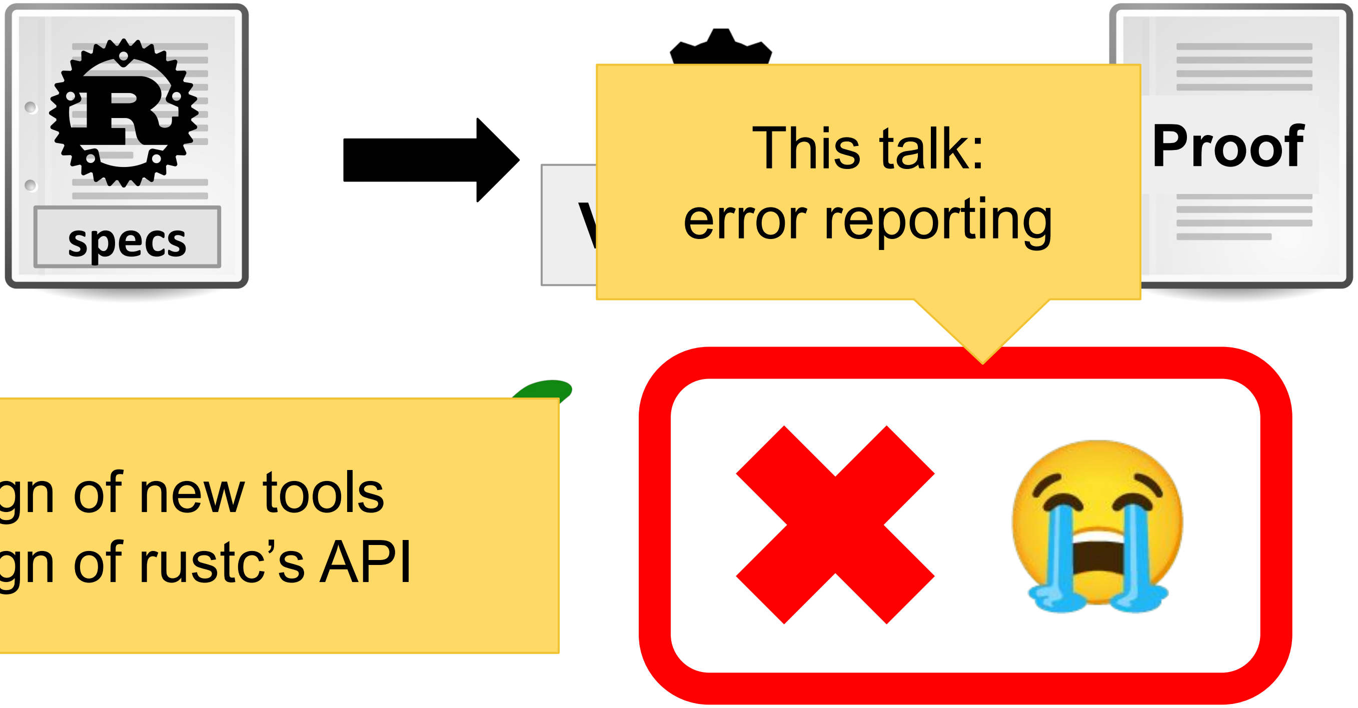
CBMC

…

Verification user experience

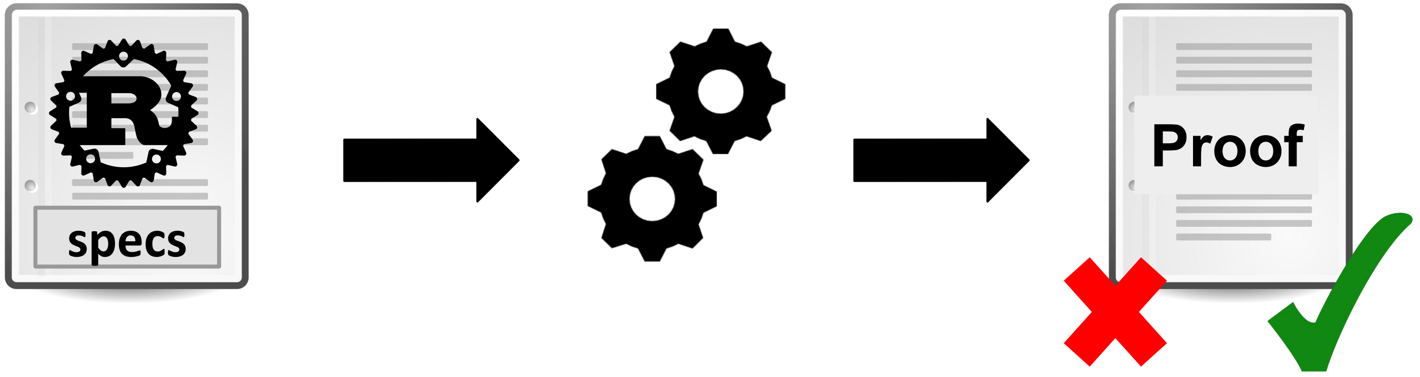


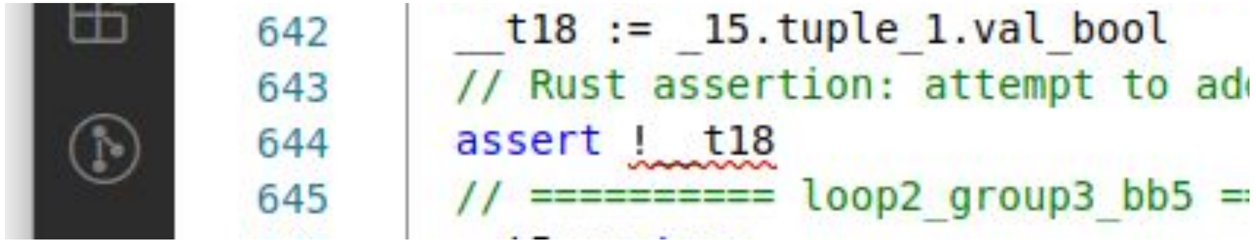
Verification user experience



- Design of new tools
- Design of rustc’s API

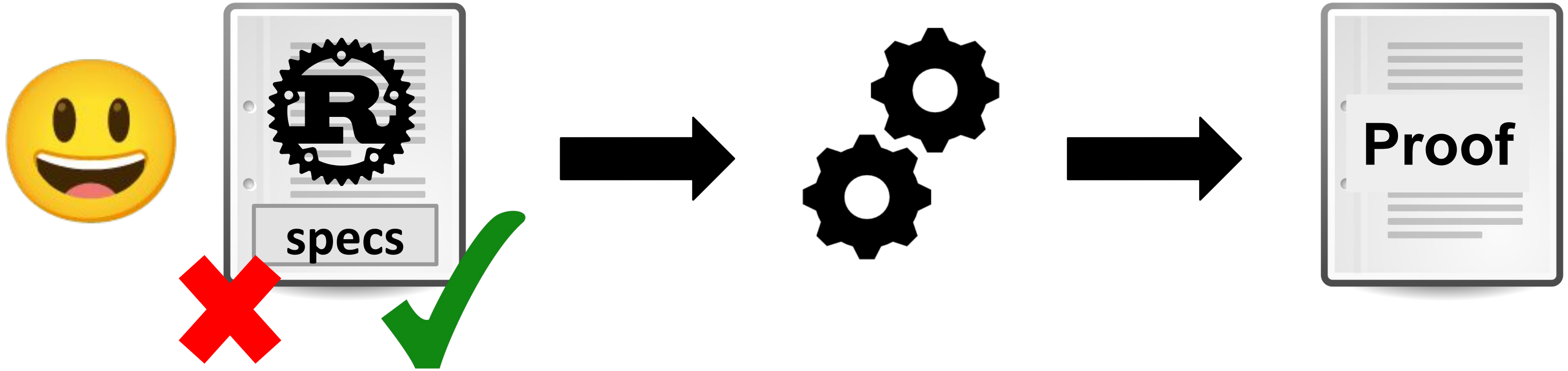
User-friendly error reporting: Proof-level?

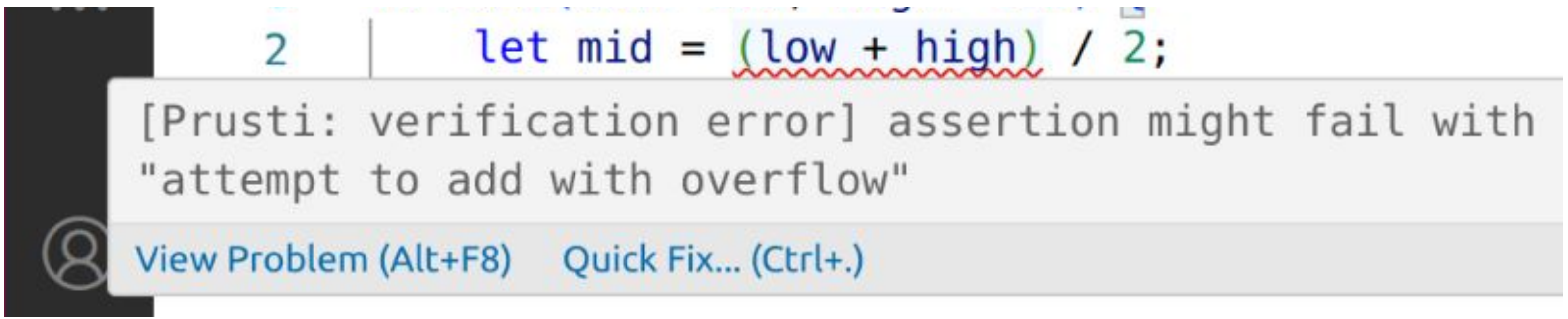




```
__t18 := _15.tuple_1.val_bool
// Rust assertion: attempt to ad
assert !__t18
// ========== loop2_group3_bb5 ==
```

User-friendly error reporting: Source-level

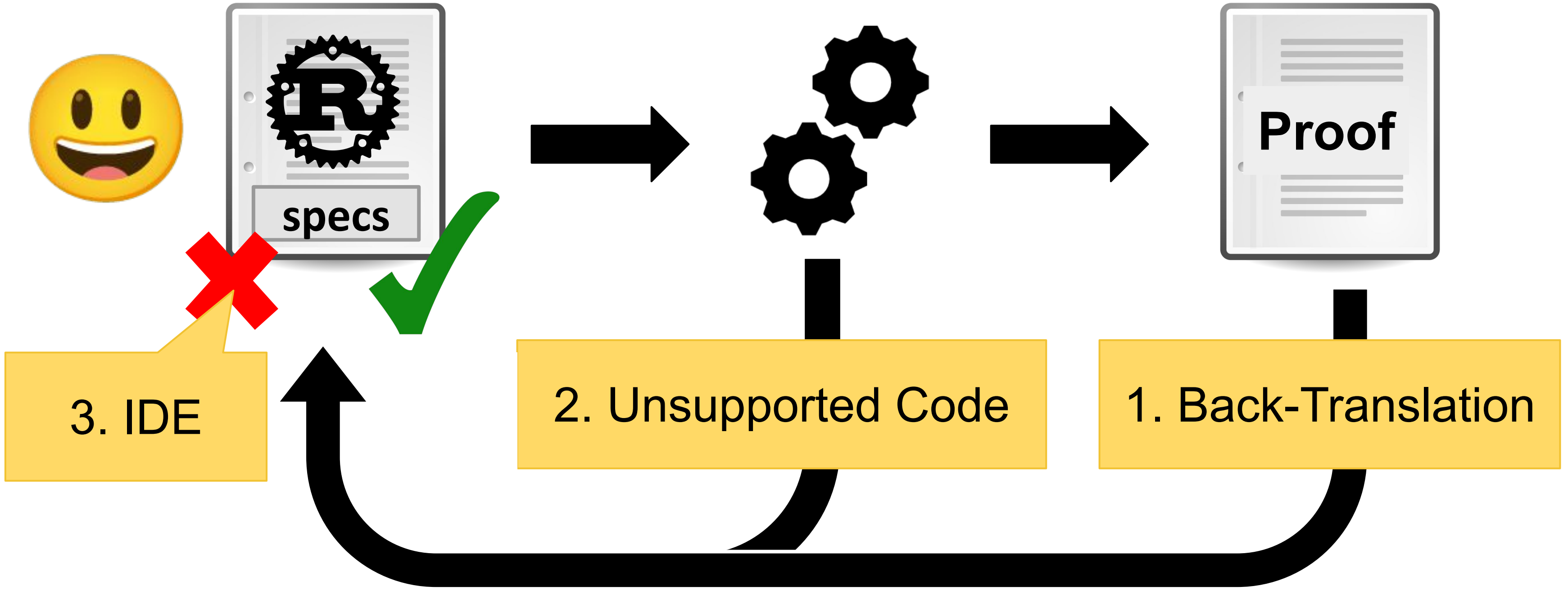




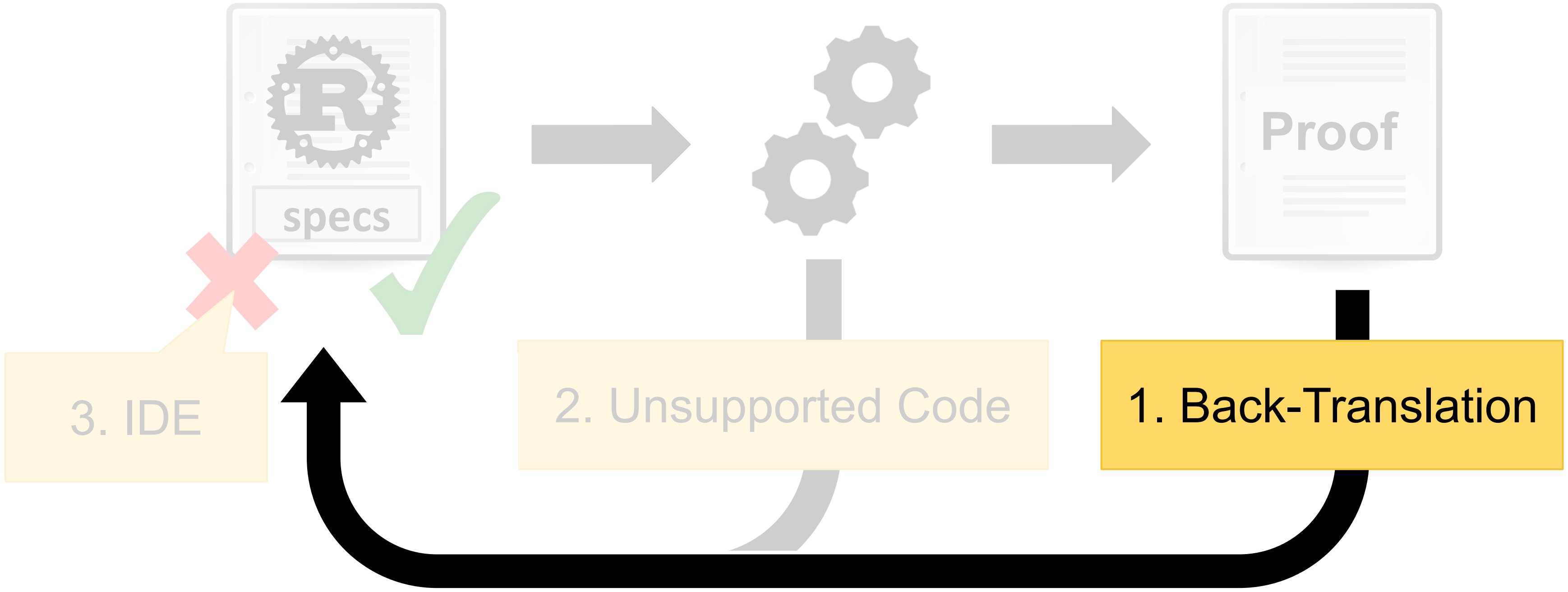
```
    let mid = (low + high) / 2;
```

[Prusti: verification error] assertion might fail with "attempt to add with overflow"

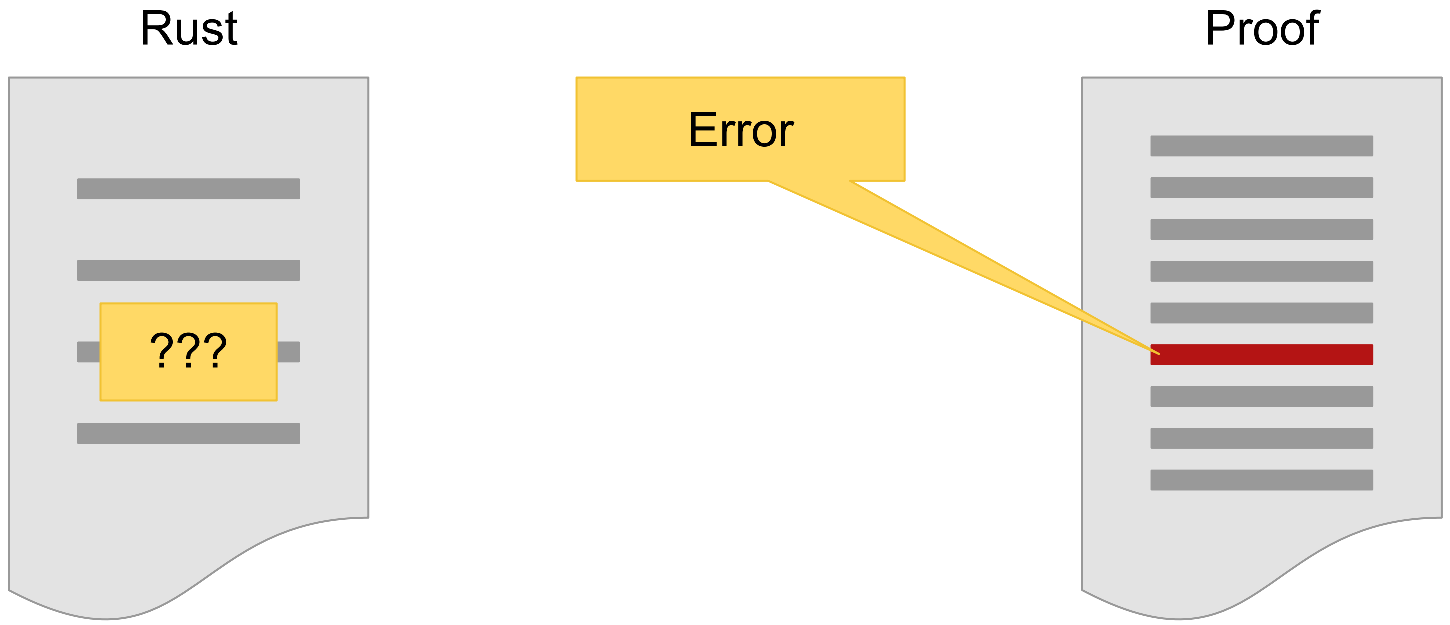
User-friendly error reporting: Source-level



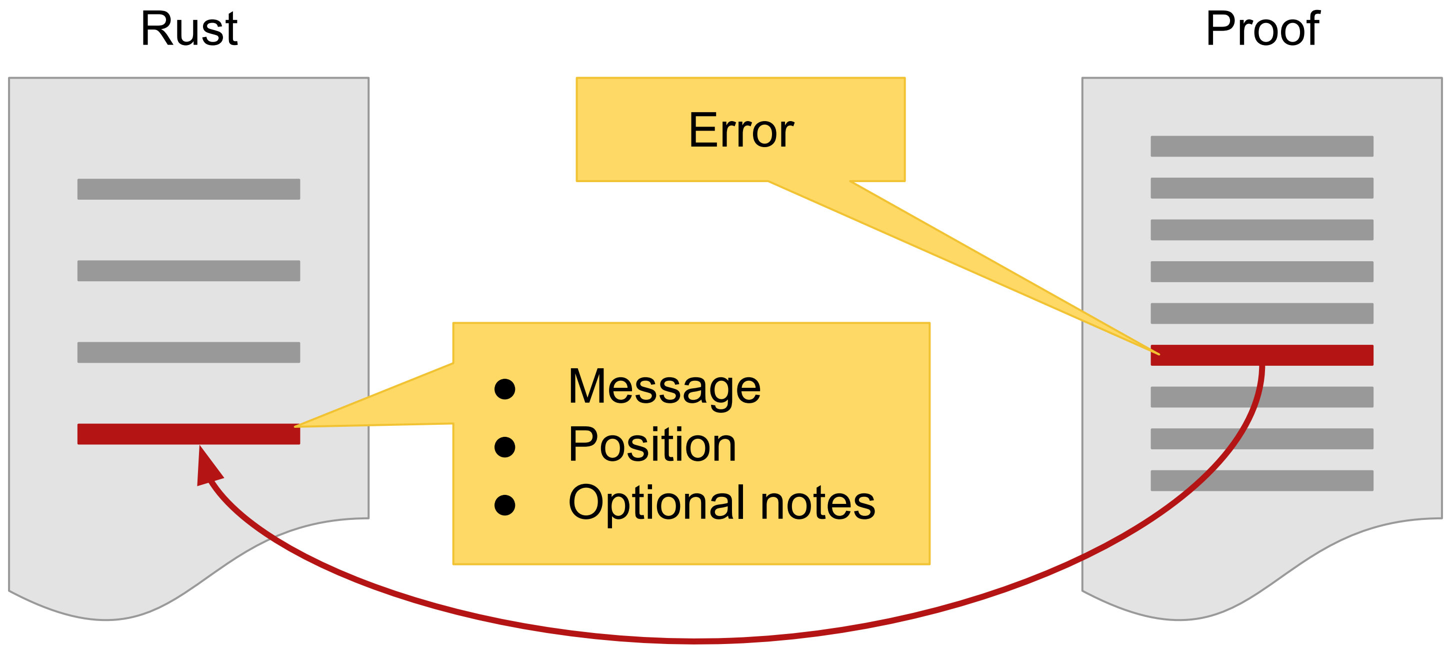
Overview



Back-translation

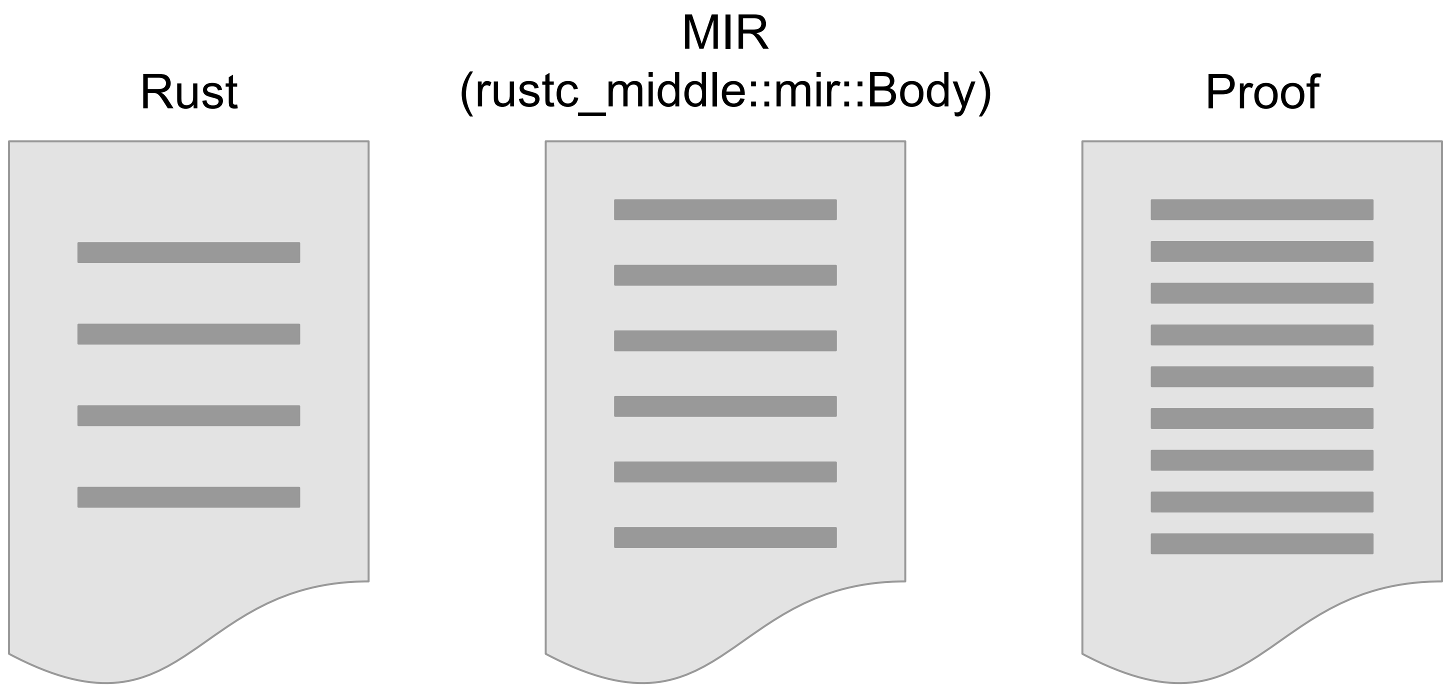


Back-translation



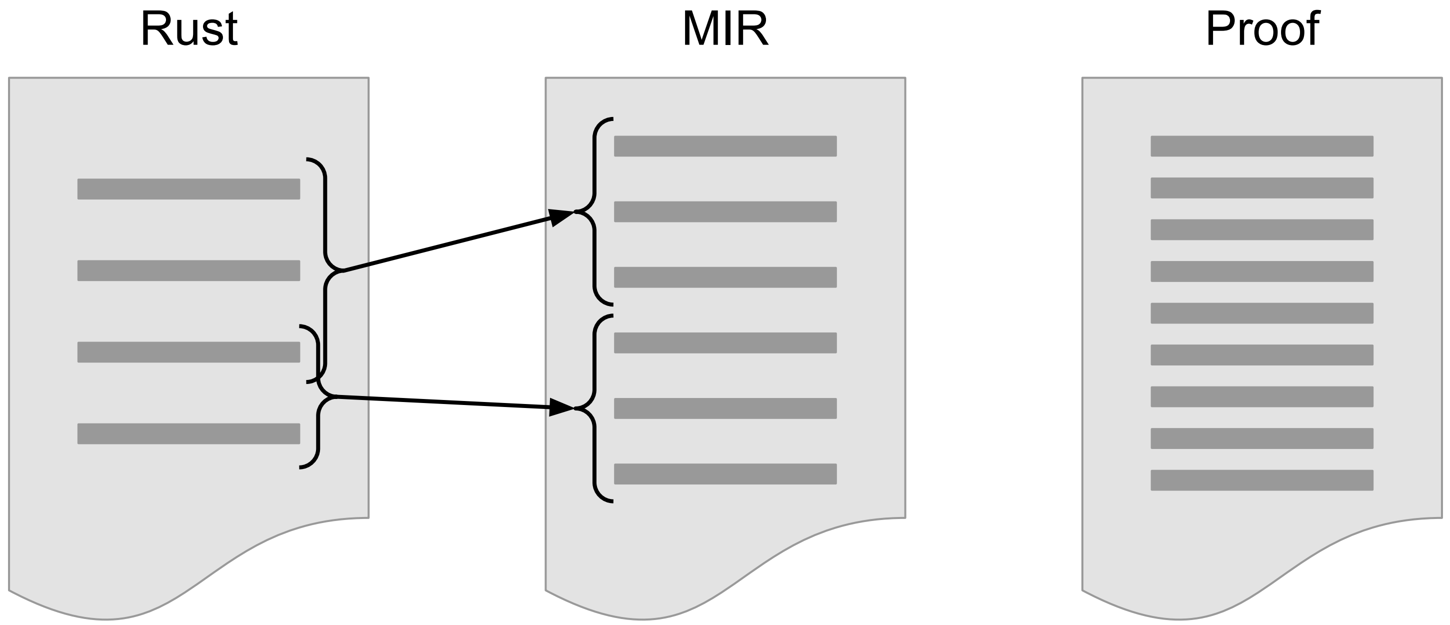
- Message
- Position
- Optional notes

MIR-based positions

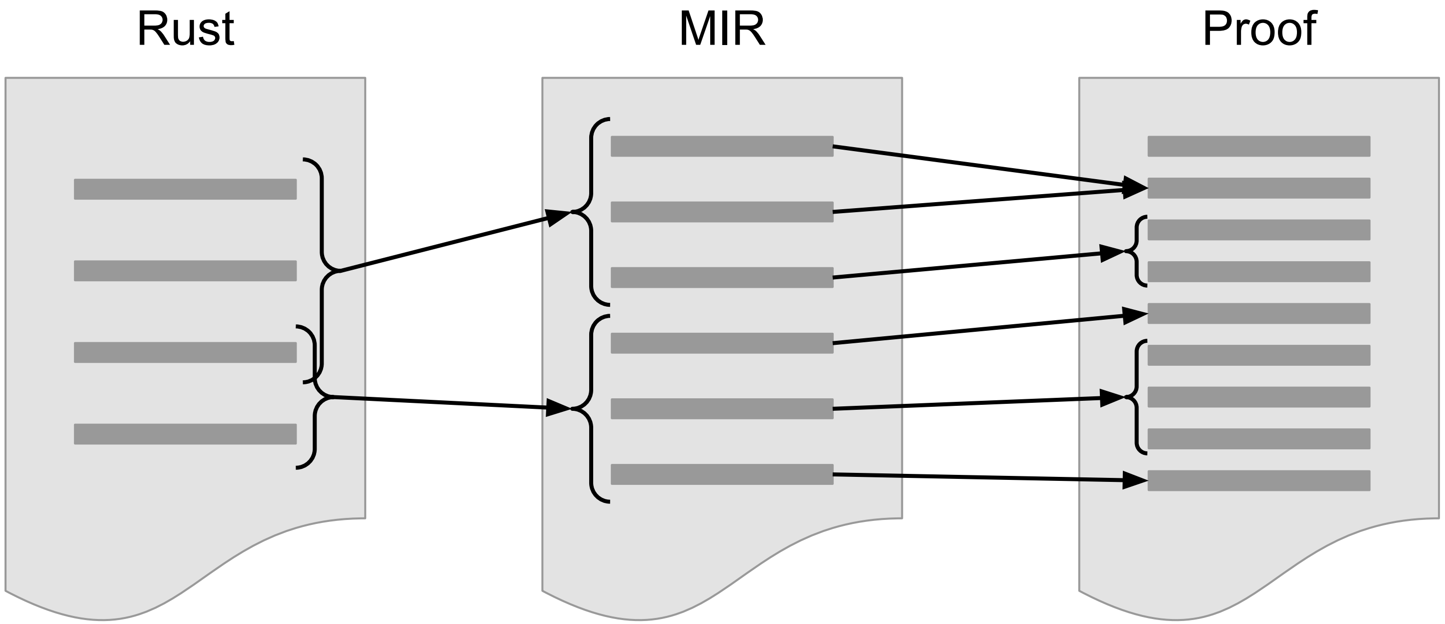


https://doc.rust-lang.org/nightly/nightly-rustc/rustc_middle/mir/struct.Body.html

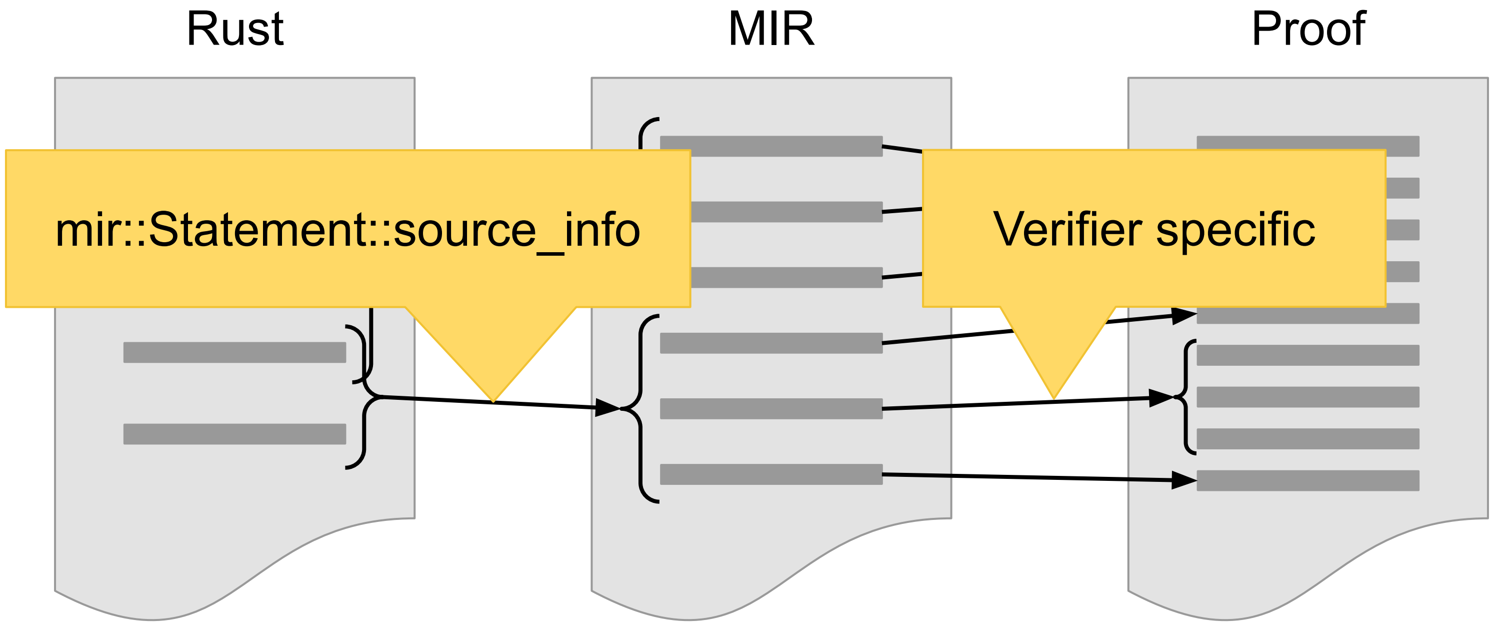
MIR-based positions



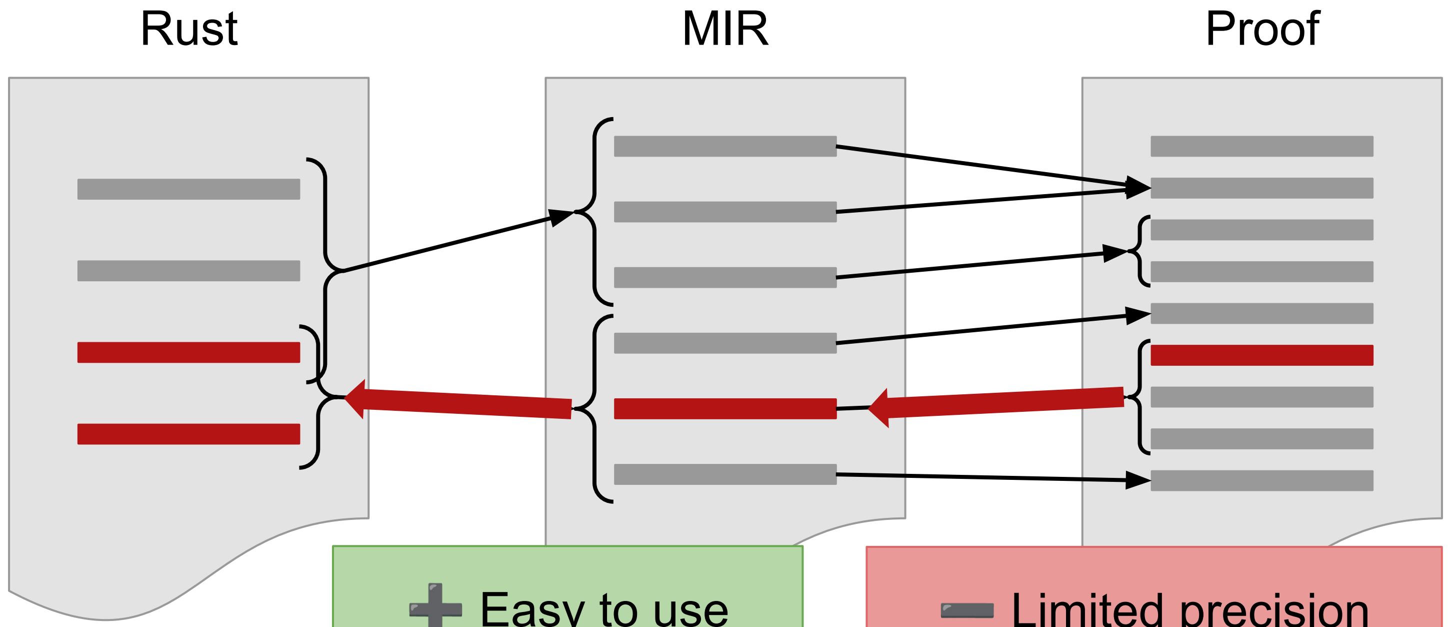
MIR-based positions



MIR-based positions



MIR-based positions



➕ Easy to use

➖ Limited precision

Limitations of MIR-based positions

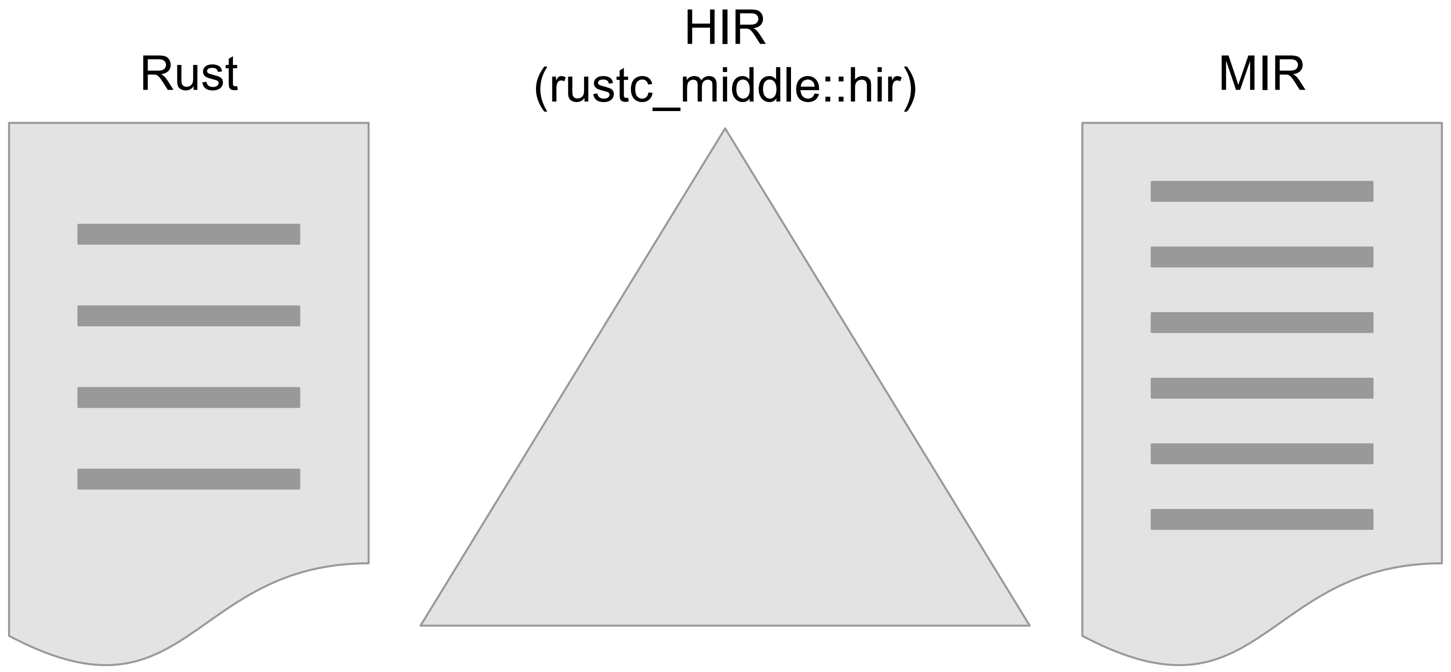
```
#[invariant(self % 2 == 0)]
type EvenU32 = u32;

fn main() {
    let mut x: EvenU32;
    x = 1;
}
```

This span is not available in MIR

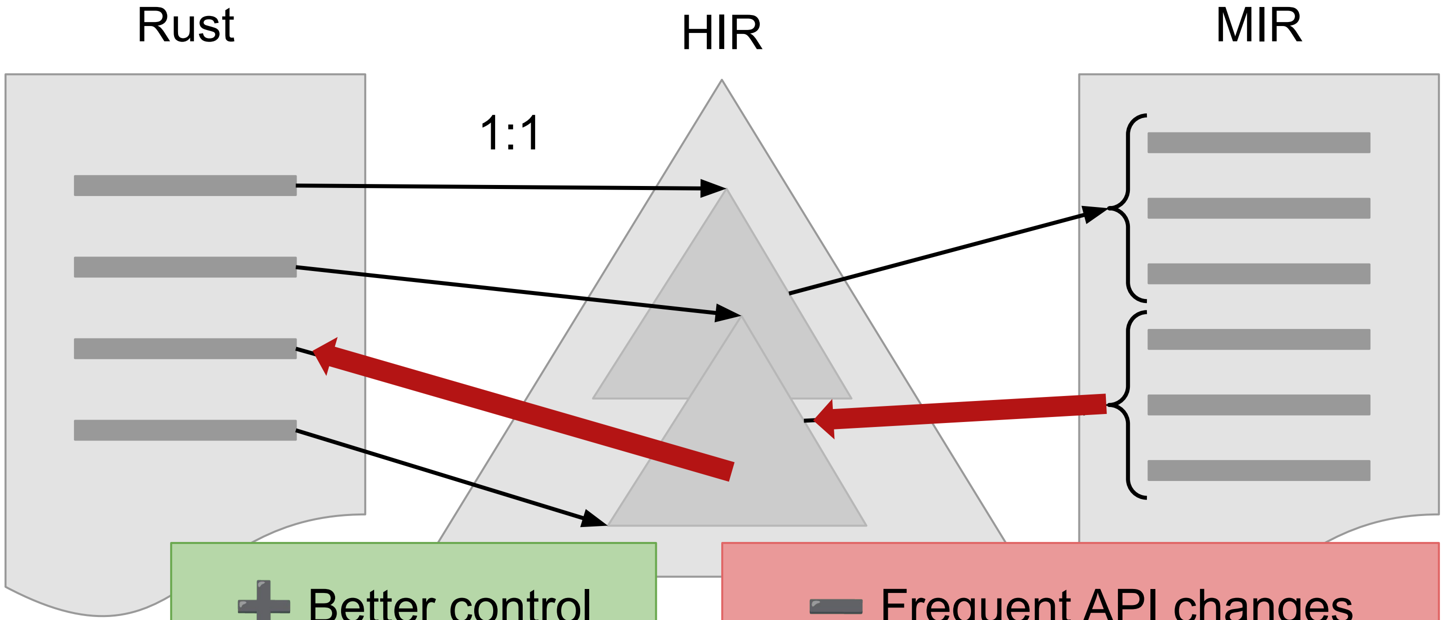
Verification error

Solution #1: HIR-based positions



https://doc.rust-lang.org/nightly/nightly-rustc/rustc_middle/hir/index.html

Solution #1: HIR-based positions



➕ Better control

➖ Frequent API changes

Solution #2: Macro-based parsing

```
your_verifier!{

   #[invariant(self % 2 == 0)]
   type EvenU32 = u32;

   fn main() {
       let mut x: EvenU32;
       x = 1;
   }

}
```

Macro

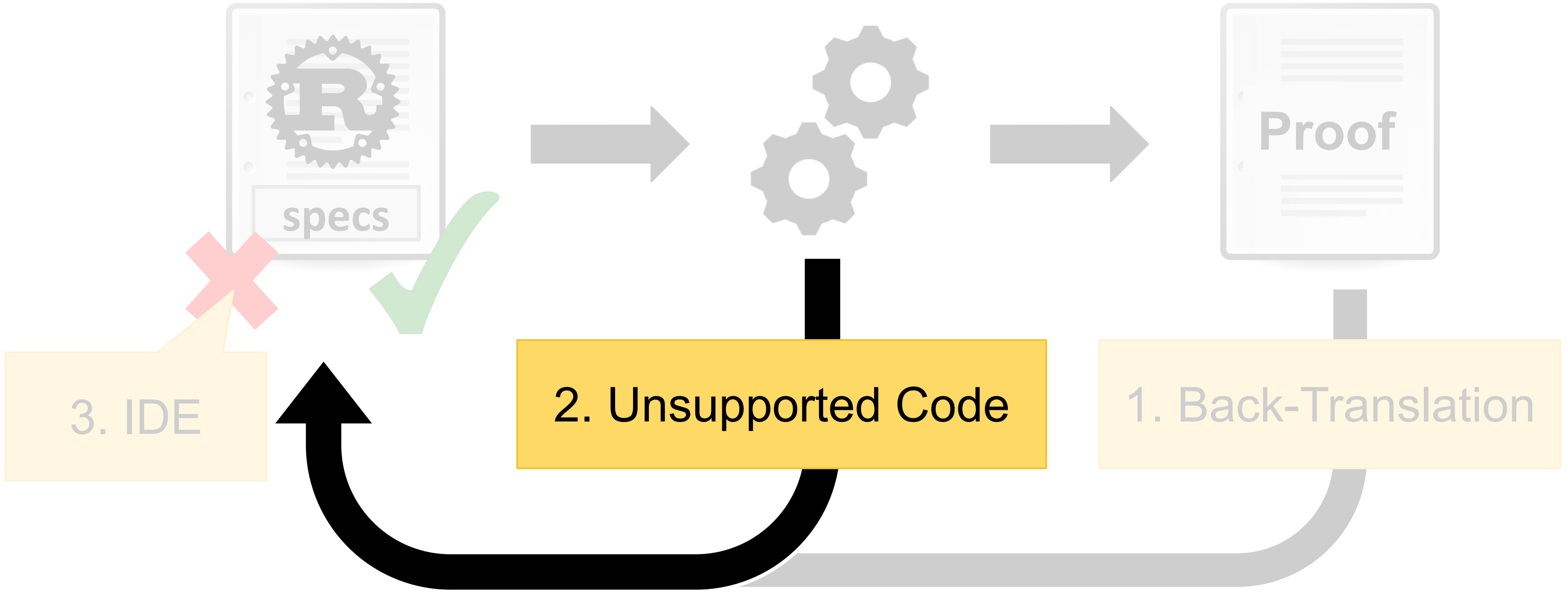
Parser: **syn** crate

➕ **Full** control

➖ Wrap code in a macro

https://crates.io/crates/syn

Overview



Usually Unsupported

Rustc feature request:
expose the drop elaboration!

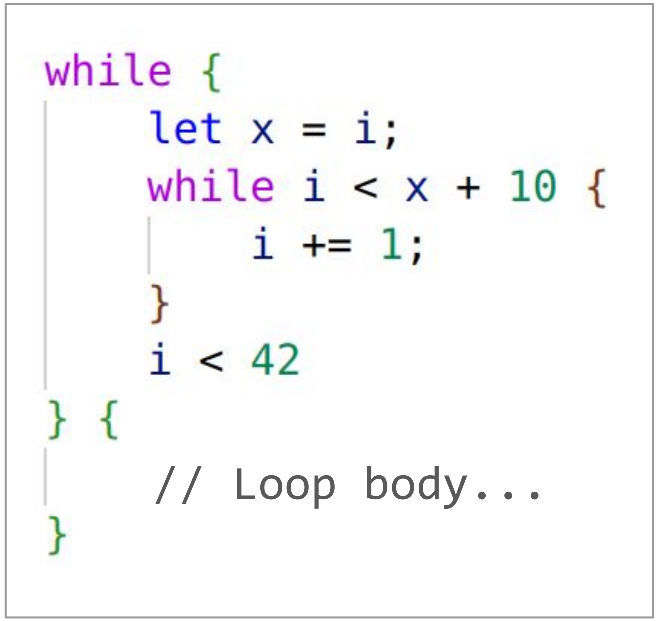
Unsafe code

Async

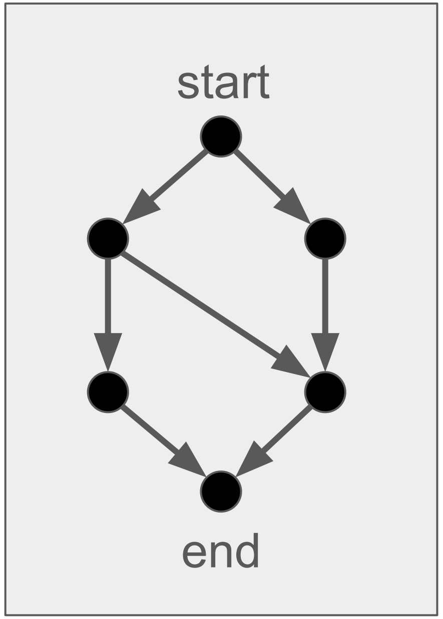
Drop

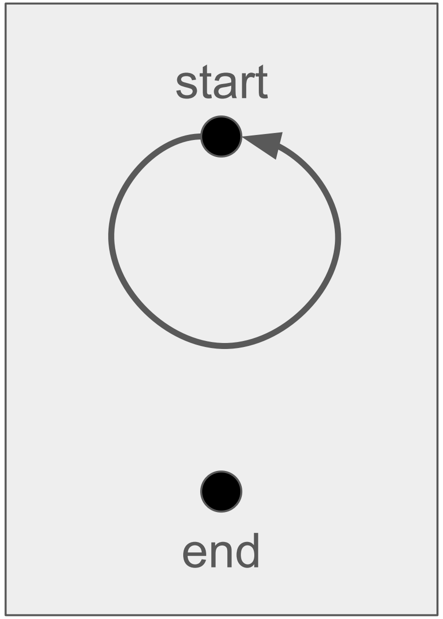
What else could be unsupported?

Strangest Rust: CFG



```rust
while {
    let x = i;
    while i < x + 10 {
        i += 1;
    }
    i < 42
} {
    // Loop body...
}
```





Strangest Rust: Types

enum with 0 variants

E.g., unsound naive axioms:

discriminant: Instance → N

discriminant(x) ∈ { v.discr for v ∈ variants(x) }

variants(x) = ∅

Strangest Rust: Types

enum with 0 variants

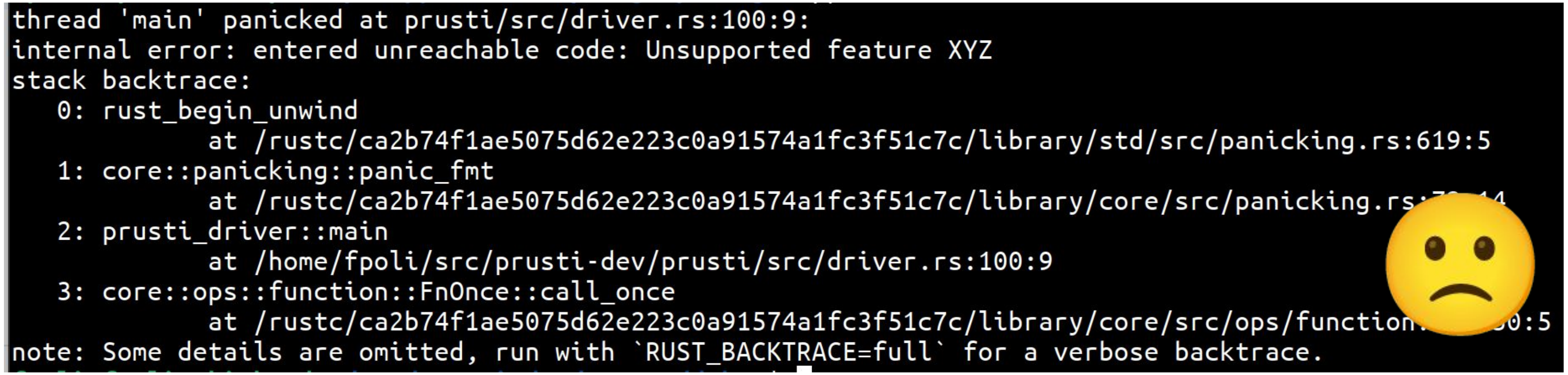
&mut ZST, **&mut impl** !Unpin

```
fn f(a: &mut ZST, b: &mut ZST) {
    let a_ptr = a as *mut _;
    let b_ptr = b as *mut _;
    assert!(a != b); // Might fail!
}
```

How to report unsupported code?

Level 1: Just Panic!

panic!(..), assert!(..), unreachable!(..), …



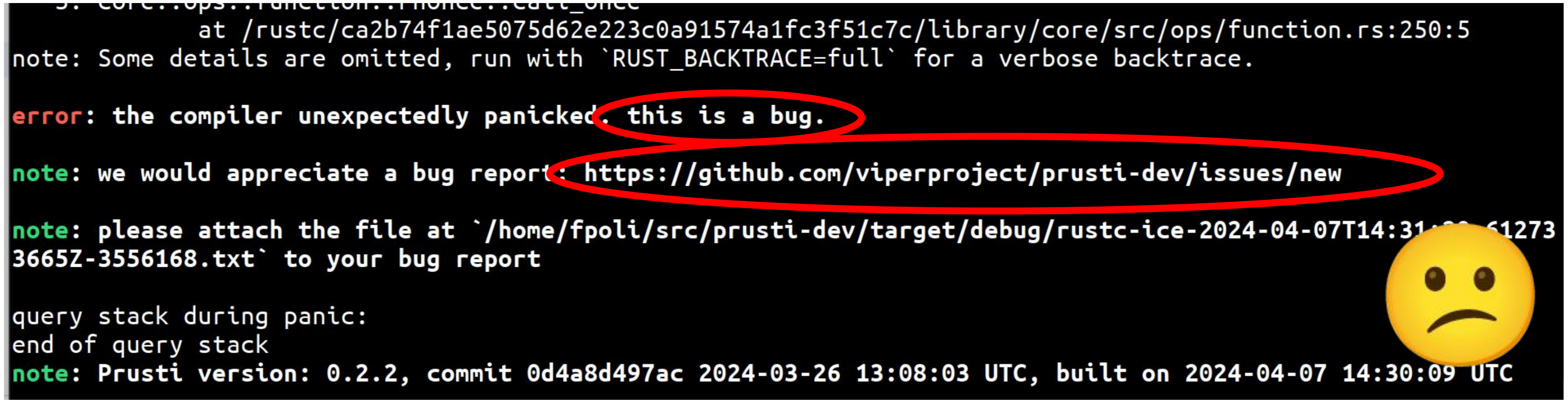
```
thread 'main' panicked at prusti/src/driver.rs:100:9:
internal error: entered unreachable code: Unsupported feature XYZ
stack backtrace:
   0: rust_begin_unwind
             at /rustc/ca2b74f1ae5075d62e223c0a91574a1fc3f51c7c/library/std/src/panicking.rs:619:5
   1: core::panicking::panic_fmt
             at /rustc/ca2b74f1ae5075d62e223c0a91574a1fc3f51c7c/library/core/src/panicking.rs[illegible]
   2: prusti_driver::main
             at /home/fpoli/src/prusti-dev/prusti/src/driver.rs:100:9
   3: core::ops::function::FnOnce::call_once
             at /rustc/ca2b74f1ae5075d62e223c0a91574a1fc3f51c7c/library/core/src/ops/function[illegible]0:5
note: Some details are omitted, run with `RUST_BACKTRACE=full` for a verbose backtrace.
```

➕ Easy to code

➖ Confusing for the user

Level 2: Set an ICE Message

rustc_driver_impl::install_ice_hook(..)



```
   3: core::ops::function::FnOnce::call_once
             at /rustc/ca2b74f1ae5075d62e223c0a91574a1fc3f51c7c/library/core/src/ops/function.rs:250:5
note: Some details are omitted, run with `RUST_BACKTRACE=full` for a verbose backtrace.

error: the compiler unexpectedly panicked. this is a bug.

note: we would appreciate a bug report: https://github.com/viperproject/prusti-dev/issues/new

note: please attach the file at `/home/fpoli/src/prusti-dev/target/debug/rustc-ice-2024-04-07T14:31[illegible]61273
3665Z-3556168.txt` to your bug report

query stack during panic:
end of query stack
note: Prusti version: 0.2.2, commit 0d4a8d497ac 2024-03-26 13:08:03 UTC, built on 2024-04-07 14:30:09 UTC
```

➕ Easy to code

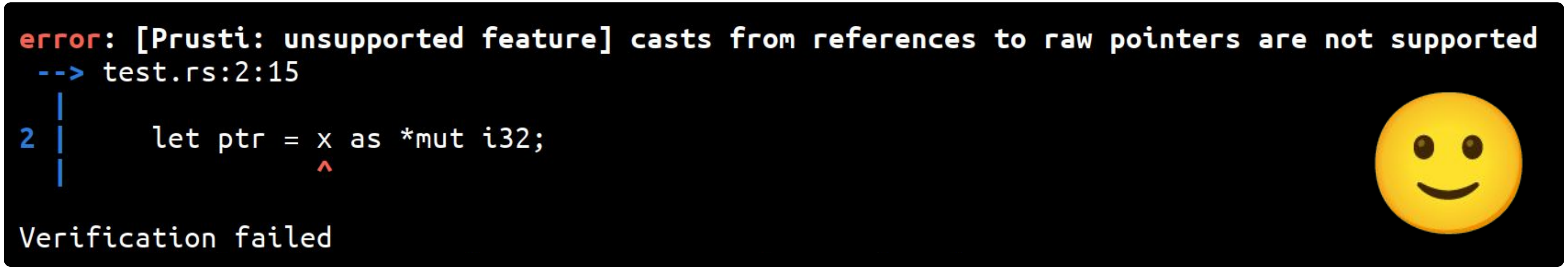
➕ Bug report link

➖ Still confusing

https://doc.rust-lang.org/nightly/nightly-rustc/rustc_driver_impl/fn.install_ice_hook.html

Level 3: Emit Rustc's Diagnostics

rustc_errors::DiagCtxt



```
error: [Prusti: unsupported feature] casts from references to raw pointers are not supported
 --> test.rs:2:15
  |
2 |     let ptr = x as *mut i32;
  |               ^

Verification failed
```

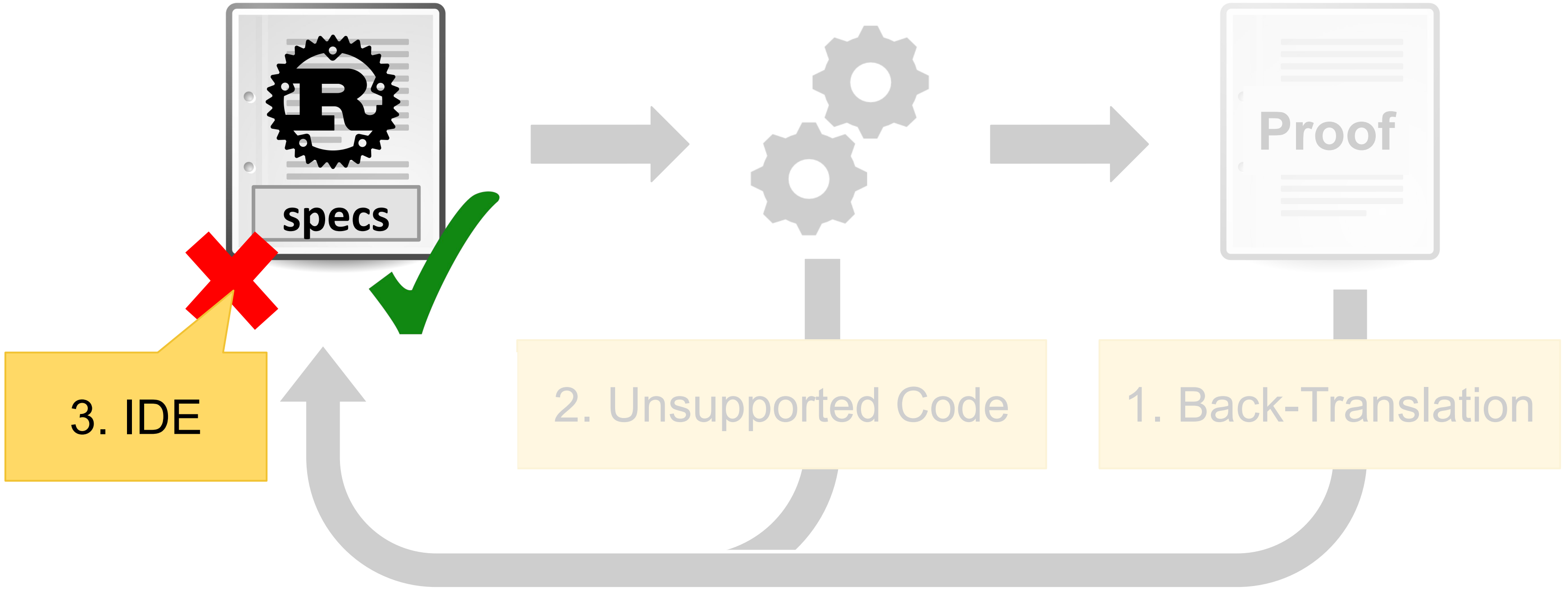
➕ Still easy to code

➕ Useful message

➕ With source-code position

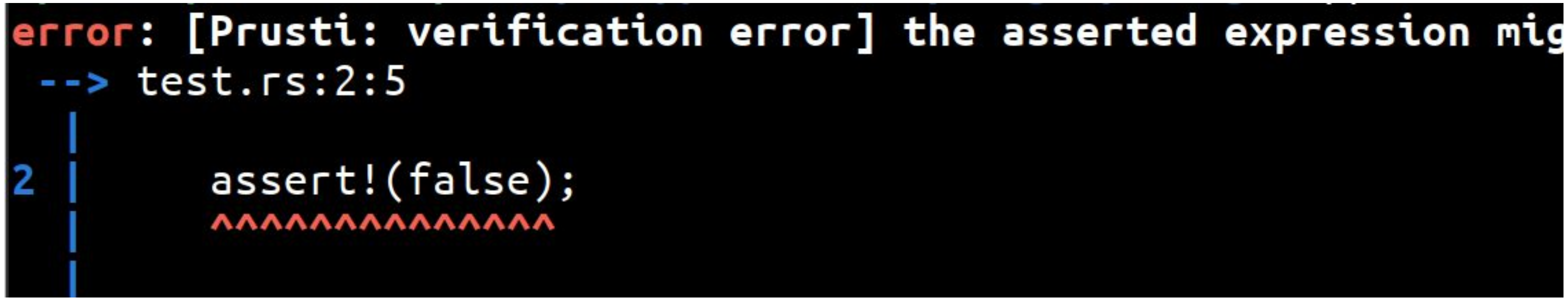
https://doc.rust-lang.org/nightly/nightly-rustc/rustc_errors/struct.DiagCtxt.html

Overview



Errors in JSON format

cargo prusti

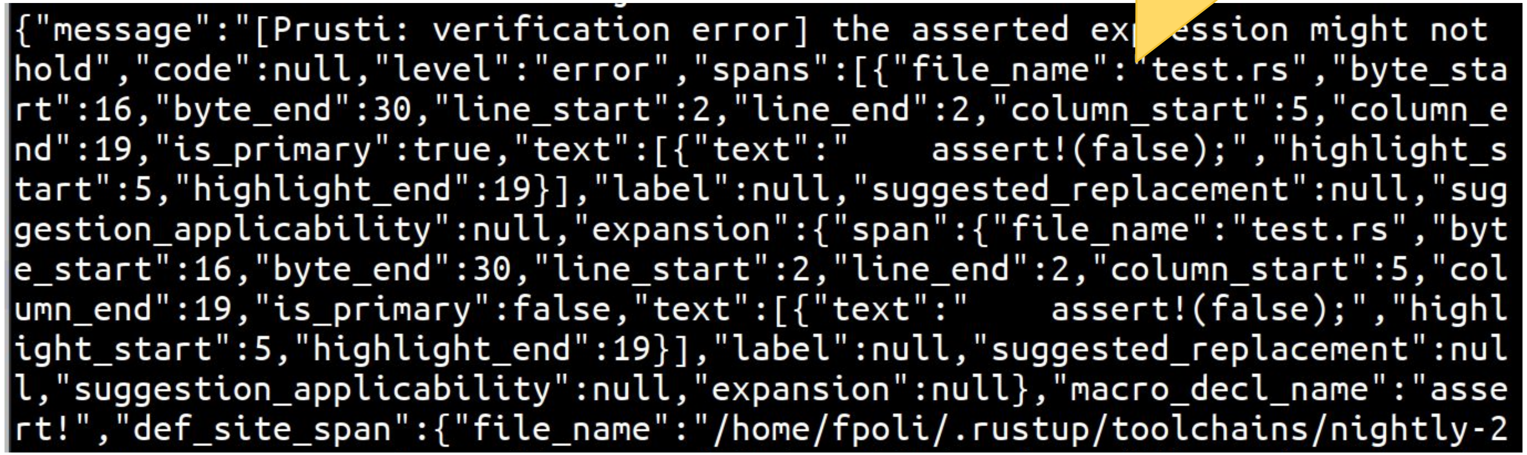


```
error: [Prusti: verification error] the asserted expression mig
  --> test.rs:2:5
   |
2  |     assert!(false);
   |     ^^^^^^^^^^^^^^
   |
```

Errors in JSON format

cargo prusti **--message-format=json**

Existing tools understand this



```
{"message":"[Prusti: verification error] the asserted ex[illegible]ession might not
hold","code":null,"level":"error","spans":[{"file_name":"test.rs","byte_sta
rt":16,"byte_end":30,"line_start":2,"line_end":2,"column_start":5,"column_e
nd":19,"is_primary":true,"text":[{"text":"    assert!(false);","highlight_s
tart":5,"highlight_end":19}],"label":null,"suggested_replacement":null,"sug
gestion_applicability":null,"expansion":{"span":{"file_name":"test.rs","byt
e_start":16,"byte_end":30,"line_start":2,"line_end":2,"column_start":5,"col
umn_end":19,"is_primary":false,"text":[{"text":"    assert!(false);","highl
ight_start":5,"highlight_end":19}],"label":null,"suggested_replacement":nul
l,"suggestion_applicability":null,"expansion":null},"macro_decl_name":"asse
rt!","def_site_span":{"file_name":"/home/fpoli/.rustup/toolchains/nightly-2
```

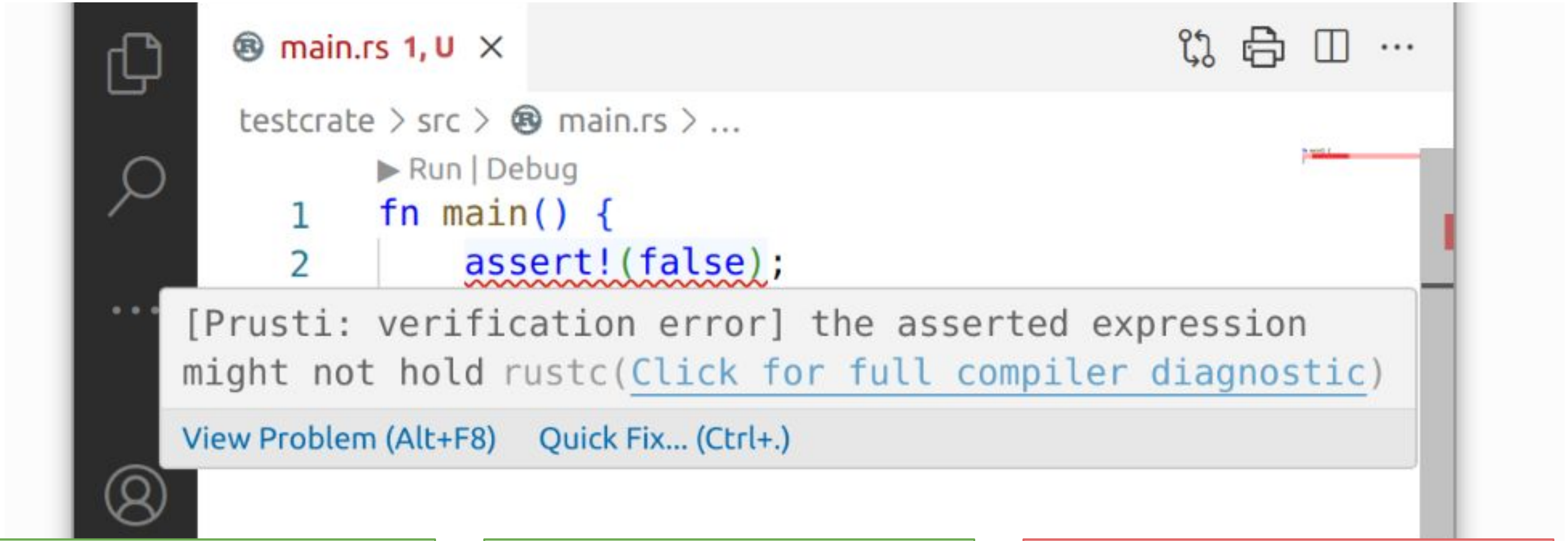
IDE integration #1: (Mis)Use Rust-Analyzer

rust.
analyzer

Settings:

```
{
        "rust-analyzer.check.overrideCommand": [
                "/path/to/cargo-prusti", "--quiet", "--workspace",
                "--message-format=json",
        ],
}
```

IDE integration #1: (Mis)Use Rust-Analyzer



```
fn main() {
    assert!(false);
```

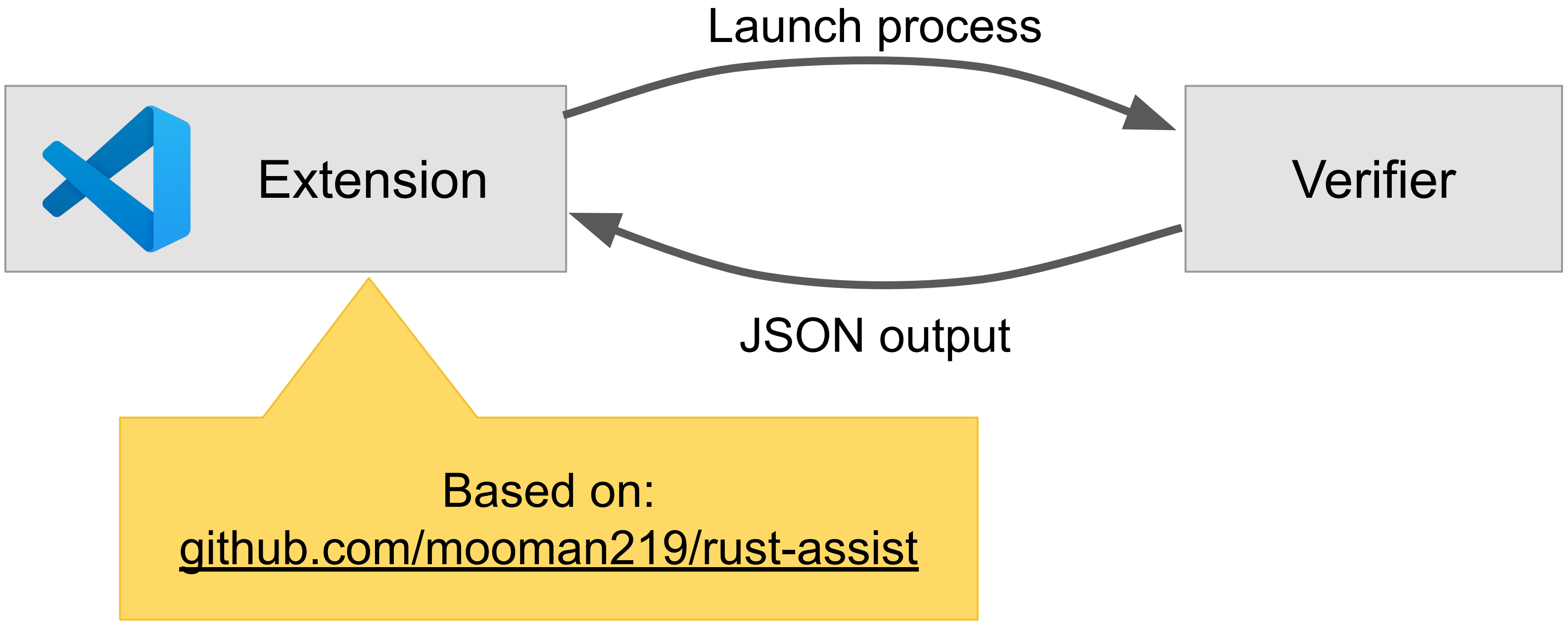
[Prusti: verification error] the asserted expression might not hold rustc(Click for full compiler diagnostic)

➕ Simple (5 min)

➕ Surprisingly good

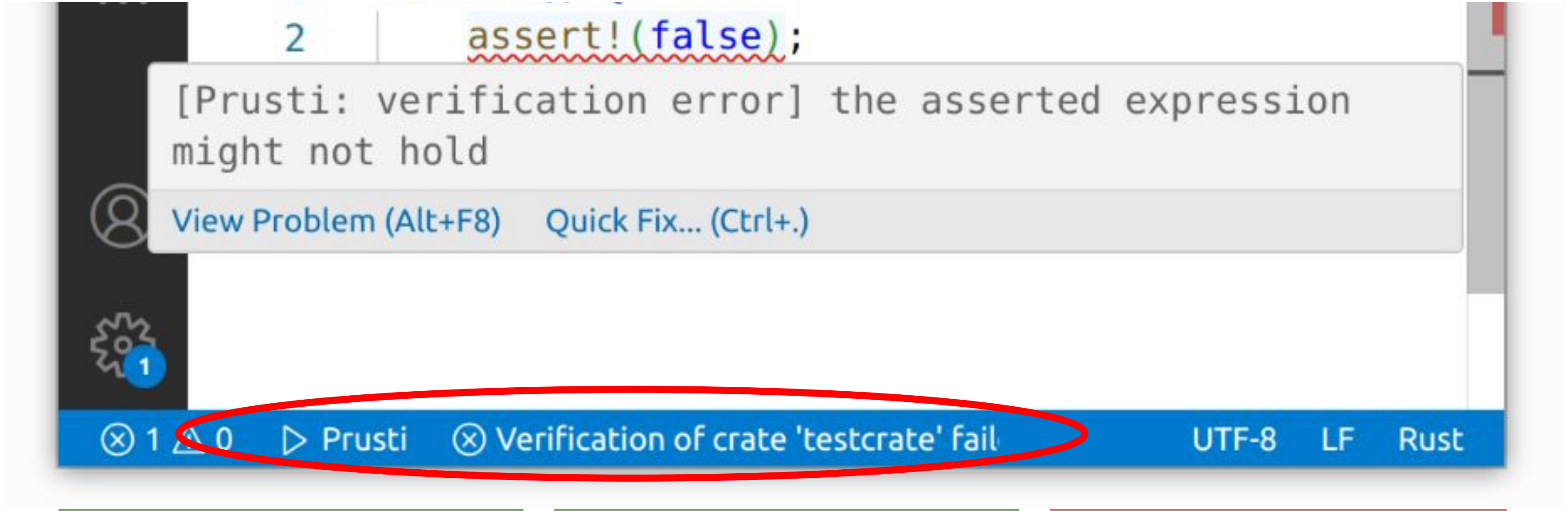
➖ Not a verifier

IDE integration #2: Parsing Rustc's JSON Diagnostics



Based on:
github.com/mooman219/rust-assist

IDE integration #2: Parsing Rustc's JSON Diagnostics



```
    assert!(false);
```

[Prusti: verification error] the asserted expression might not hold

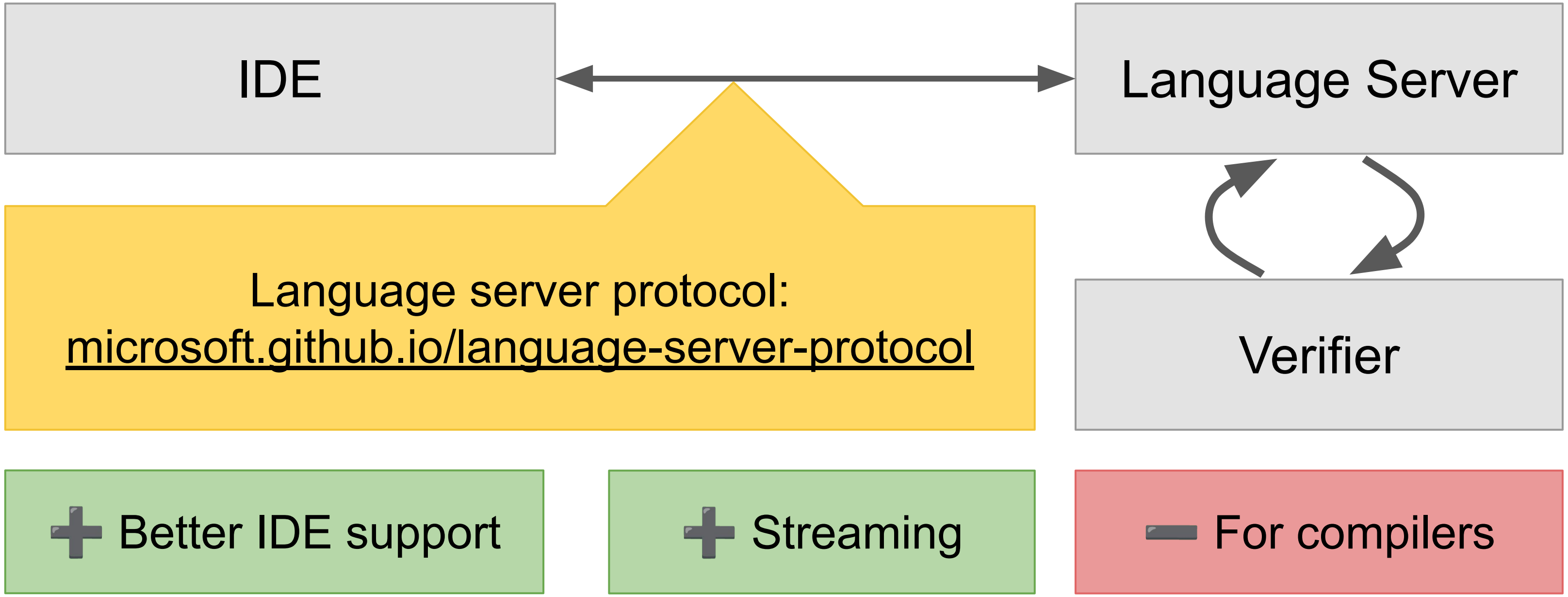
⊗ Verification of crate 'testcrate' fail

➕ Simple (~days)

➕ Easy to customize

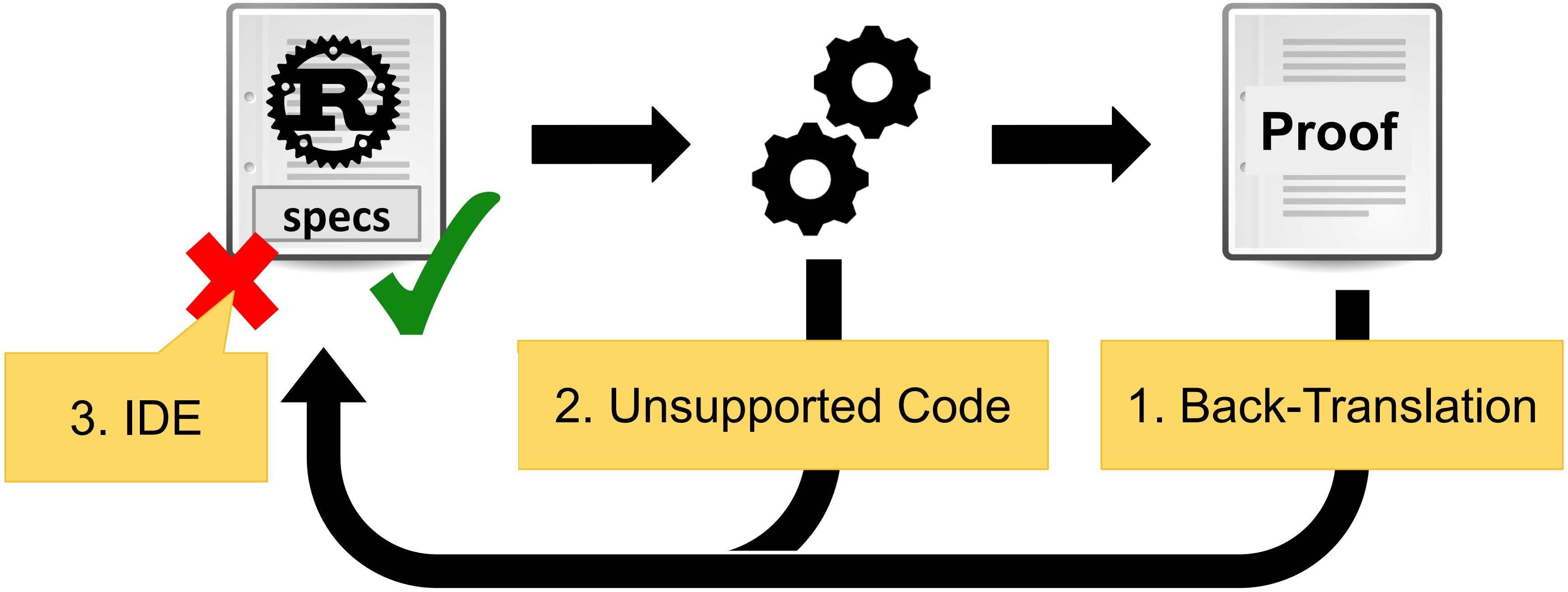
➖ One-way cannel

IDE integration #3: Language Server Protocol



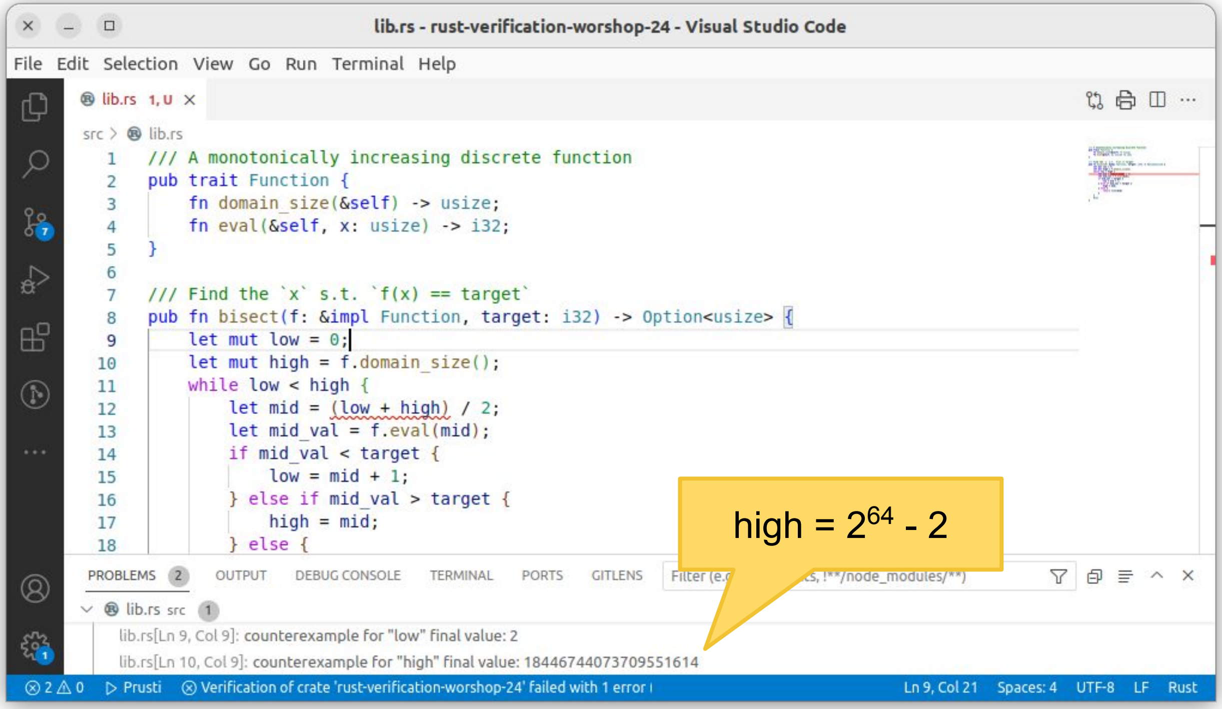
Thank you!

🔗 github.com/viperproject/prusti-dev



Backup Slides

Error reporting with counterexamples


lib.rs - rust-verification-worshop-24 - Visual Studio Code
File Edit Selection View Go Run Terminal Help
lib.rs 1, U
src > lib.rs
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
Filter (e.g. ...ts, !**/node_modules/**)
lib.rs src 1
⊗ 2 ⚠ 0 Prusti
Ln 9, Col 21 Spaces: 4 UTF-8 LF Rust

```rust
1   /// A monotonically increasing discrete function
2   pub trait Function {
3       fn domain_size(&self) -> usize;
4       fn eval(&self, x: usize) -> i32;
5   }
6
7   /// Find the `x` s.t. `f(x) == target`
8   pub fn bisect(f: &impl Function, target: i32) -> Option<usize> {
9       let mut low = 0;
10      let mut high = f.domain_size();
11      while low < high {
12          let mid = (low + high) / 2;
13          let mid_val = f.eval(mid);
14          if mid_val < target {
15              low = mid + 1;
16          } else if mid_val > target {
17              high = mid;
18          } else {
```

```
lib.rs[Ln 9, Col 9]: counterexample for "low" final value: 2
lib.rs[Ln 10, Col 9]: counterexample for "high" final value: 18446744073709551614
```

⊗ Verification of crate 'rust-verification-worshop-24' failed with 1 error

high = $2^{64} - 2$