Rust 代码分析实践 🤔

🤔 0 -> 1 😄

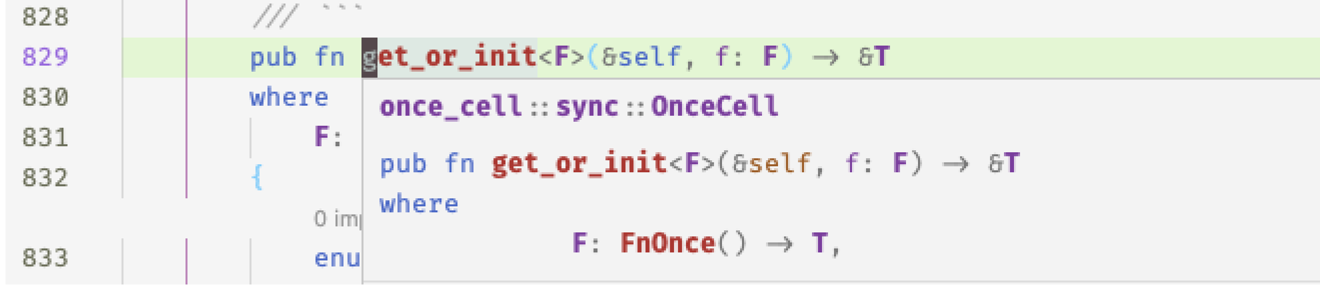
Versions:
rustc: 1.45.0-nightly (1836e3b42 2020-05-06)
rust-analyzer: bb697727

2020-12-27 尹思维 from Feishu Team

代码分析?

- 案例
  - clippy: a collection of lints.
  - miri: detect undefined behavior.
  - rust-analyzer: provide IDE features.

- 为什么要定制 ? 🤔
  - 缺失 Workspace 级别的检查，如 Workspace 级别的 Dead Code 。
  - 难以实现增量检查。
  - …



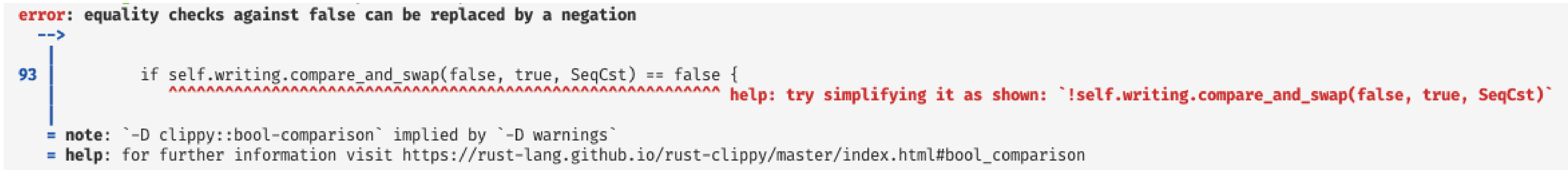
```
/// ```
pub fn get_or_init<F>(&self, f: F) -> &T
where
    F:
{
    0 im
    enu
```

```
once_cell::sync::OnceCell

pub fn get_or_init<F>(&self, f: F) -> &T
where
        F: FnOnce() -> T,
```

👆 Hover 😎

👇 Clippy 😎

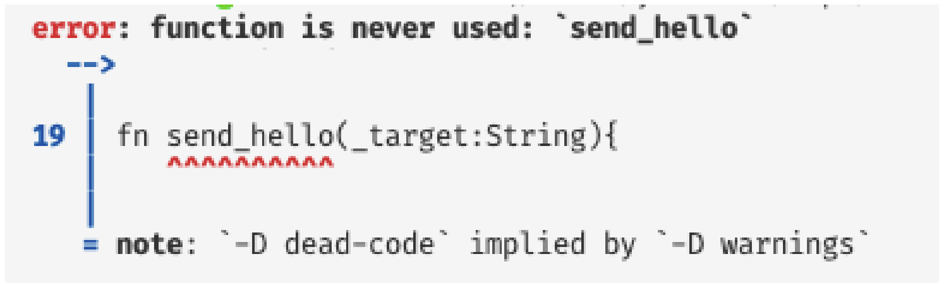


```
error: equality checks against false can be replaced by a negation
  -->
   |
93 |         if self.writing.compare_and_swap(false, true, SeqCst) == false {
   |            ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^ help: try simplifying it as shown: `!self.writing.compare_and_swap(false, true, SeqCst)`
   |
   = note: `-D clippy::bool-comparison` implied by `-D warnings`
   = help: for further information visit https://rust-lang.github.io/rust-clippy/master/index.html#bool_comparison
```

Workspace-level unused pub

```
project/crates

crate_a
crate_b
crate_...
```



```
error: function is never used: `send_hello`
  -->
   |
19 | fn send_hello(_target:String){
   |    ^^^^^^^^^^^
   |
   = note: `-D dead-code` implied by `-D warnings`
```

unused 💔

unused 💔

✓ used by other crates

```
// crate_a

pub fn send_text(text: String) { /* ... */ }

pub fn send_card(card: Card) { /* ... */ }

pub fn send_message(msg: Message) { /* ... */ }
```

Incremental SQL check

```
dsl::settings_fields    .select(dsl::value)    .filt
er(dsl::key.eq(field_name))    .first::<String>(
conn)
```

explain query plan select value from settings_fields where key = ?;

```
QUERY PLAN
`--SEARCH TABLE settings_fields USING PRIMARY 😀KEY    (key=?)
                                 USING TEMP B-TREE 😠
```

定制 Clippy: 每次 Clippy check 的时候都会检查全部的 Diesel Query DSL 。

期望的效果 ： 当且仅当 Diesel Query DSL 改变的时候。

Simple methods

ripgrep (or other grep tools) ~ 0.12s for 3000 .rs files 😎

```
rg  -g '*.rs' 'Regex::'
```

syn ~ 37.68s for 3000 .rs files

```
use std::{fs::File, io::Read};
fn main() {
    glob::glob("**/*.rs")
     .unwrap()
     .filter_map(Result::ok)


        .for_each(|path| {
        let mut content = String::new();
        File::open(path).unwrap().read_to_string(&mut content).unwrap();
        let _ = syn::parse_file(&content);
     });
}
```

syn + rayon ~ 14.58s for 3000 .rs files 😎😎

Simple example

```rust
pub mod example {
    pub mod inner {
        pub fn some_func(){}
    }
    mod inner2 {
        pub fn some_func2(){}
    }
}
```

- 一个可以扫描所有公开函数，
  并输出其全限定名称的小工具。
- A tool scans all **public** function declarations and prints their signatures **with fully-qualified names.**

```
rg  -g '*.rs' 'pub fn'
```

Result:

```
pub fn some_func(){}
pub fn some_func2(){}
```

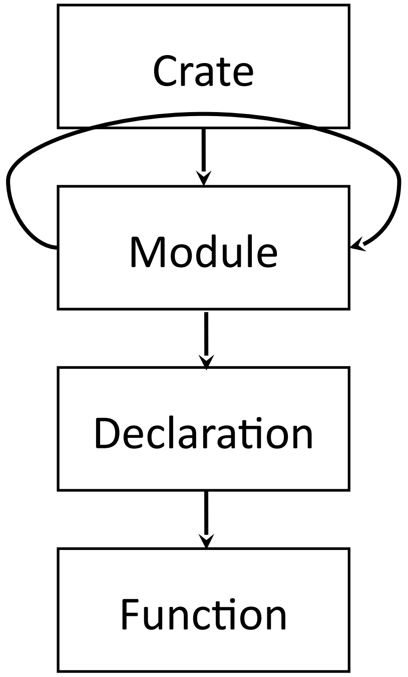
Expect:

```
example::inner::some_func()
```

*`some_func2` 在非公开 mod `inner2` 中被定义，所以不是对外可见的。

Simple example / How-to?

```rust
pub mod example {
    pub mod inner {
        pub fn some_func(){}
      }
    mod inner2 {
        pub fn some_func2(){}
    }
}
```



1. 从一个 crate 的根 mod 开始。
2. 输出所有在该 mod 中定义的公开函数签名。
3. 遍历所有公开的子 mod ，依次执行 #2 。

Have fun with rust-analyzer 😎😎😎

Add Dependencies

```toml
[dependencies]
rust-analyzer = { git = "https://github.com/rust-analyzer/rust-analyzer.git" }
ide = { git = "https://github.com/rust-analyzer/rust-analyzer.git" }
hir = { git = "https://github.com/rust-analyzer/rust-analyzer.git" }
```

Bootstrap

```rust
let args = env::args().skip(1).take(1).next();
let prj_dir = args.expect("project dir must be specified.");
let (host, _vfs) = rust_analyzer::cli::load_cargo(prj_dir.as_ref(), true, false).unwrap();
let db: &ide::RootDatabase = host.raw_database();
for krate in hir::Crate::all(db) {
    println!("Found crate {}", krate.display_name(db).unwrap());
}
```

Run

```
cargo run -- .

Prepare to scan ..
Found crate alloc
Found crate generator
..
```
~ 600 crate names are printed

Structure

```
rust-analyzer/crates

- rust-analyzer   -> main entry
- ide, ide_db     -> IDE programmable API
- parser          -> source_code -> Token
- syntax          -> Token -> AST
- hir, hir_def,...-> semantic information
```

RA: API overview (Top-Down)

AnalysHost

```
let (host, vfs) = rust_analyzer::cli::load_cargo(...).unwrap();
```

RootDatabase

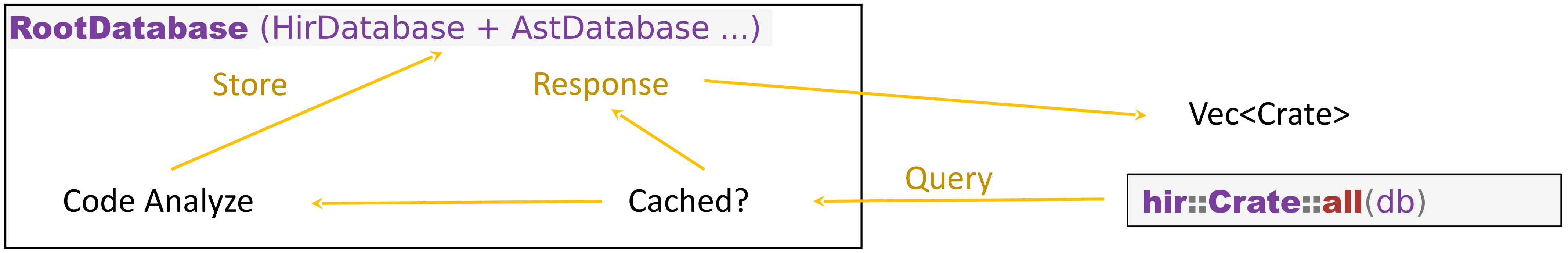
```
let db: &ide::RootDatabase = host.raw_database();
```

Crate

```
let krate: hir::Crate = hir::Crate::all(db)[0];
```

Module

```
let module: hir::Module = krate.root_module(db);
```



RA: API overview (Top-Down)

Declations

```
let decls: hir::ModuleDef = module.declarations(db)[0];
```

Function

```
let func: hir::Function = todo!();
 let name = func.name(db);
let params = func.method_params(db);
```

Back to source code

```
// AST Node
let func: syntax::ast::Fn = func.source(db).value;
// Token
let fn_token: syntax::SyntaxToken = func
      .fn_token().unwrap();
// Source code offset
let offset = fn_token.text_range().start();
```

```
pub enum ModuleDef {
    Module(Module),
    Function(Function),
    Adt(Adt),
    Trait(Trait),
    ...
}

pub enum Adt {
       Struct(Struct),
       Union(Union),
       Enum(Enum),
}
```

```
rustc_driver::run_compiler(&env::args().collect::<Vec<_>>(), &mut Cb {}, None, None)
```

Have fun with rustc 😎😎😎

```
#![feature(rustc_private)]
extern crate rustc_driver;
extern crate rustc_interface;
extern crate rustc_hir;
extern crate rustc_middle;
struct Cb {}
impl rustc_driver::Callbacks for Cb {
    fn after_parsing<'tcx>(
    fn after_expansion<'tcx>(
```

- rustc_driver
  用于调用编译过程
- rustc_interface
  用于从编译器中获取信息

  - after_parsing
  - after_expansion
  - after_analysis

```
   fn after_analysis<'tcx>(
      &mut self,
      compiler: &rustc_interface::interface::Compiler,
      queries: &'tcx rustc_interface::Queries<'tcx>,
   -> Compilation {
      queries.global_ctxt().unwrap().peek_mut().enter(|tcx| { // do sth with tcx  });
      Compilation::Continue
   }
}
```

Have fun with rustc & Cargo

- RUSTC_WRAPPER

```
RUSTC_WRAPPER=<your rustc> cargo check
```

- 有用的信息

Clippy: https://github.com/rust-lang/rust-clippy/blob/master/src/driver.rs

Miri: https://github.com/rust-lang/miri/blob/master/src/bin/miri.rs

RUSTC: API overview (Top-Down)

```
queries.global_ctxt().unwrap().peek_mut().enter(|tcx| {
```

tcx: rustc_middle::ty::context::TyCtxt

HirMap: rustc_middle::hir::map::Map

```
let h = tcx.hir();
```

Crate: rustc_hir::hir::Crate

```
let krate = h.krate();
```

Item: rustc_hir::hir::Item

```
for (_, item) in &krate.items {
    if let rustc_hir::ItemKind::Fn(ref sig, _, body_id) = item.kind {
      let local_def_id = tcx.hir().body_owner_def_id(body_id);
```

mir: rustc_middle::mir::Body

```
      let mir = tcx.optimized_mir(local_def_id.to_def_id());
      let param_env = tcx.param_env(local_def_id.to_def_id());

     //..
}
```

RUSTC vs RA

```
struct Item<'hir> {
    ident: Ident
    hir_id: HirId
    attrs: &'hir [Attribute]
    kind: ItemKind<'hir>
    vis: Visibility<'hir>
    span: Span
}
```

```
pub enum ItemKind<'hir> {
    ExternCrate(Option<Symbol>),
    Use(&'hir Path<'hir>, UseKind),
    Static(&'hir Ty<'hir>, Mutability, BodyId),
    Const(&'hir Ty<'hir>, BodyId),
    Fn(FnSig<'hir>, Generics<'hir>, BodyId),
    Mod(Mod<'hir>),
    ForeignMod {...
    GlobalAsm(&'hir GlobalAsm),
    TyAlias(&'hir Ty<'hir>, Generics<'hir>),
    OpaqueTy(OpaqueTy<'hir>),
    Enum(EnumDef<'hir>, Generics<'hir>),
    Struct(VariantData<'hir>, Generics<'hir>),
    Union(VariantData<'hir>, Generics<'hir>),
    Trait(..
    Impl {...}
```

RUSTC

```
pub enum ModuleDef {
    Module(Module),
    Function(Function),
    Adt(Adt),
    Trait(Trait),
    ...
}

pub enum Adt {
     Struct(Struct),
     Union(Union),
     Enum(Enum),
}
```

RA

Basic block

```
w = 0;
x = x + y;
y = 0;
if (x > z)
{
    y = x;
    x++;
}
else
{
    y = z;
    z++;
}
w = x + z;
```

Source
源代码

```
w = 0;
x = x + y;
y = 0;
if (x > z)
```

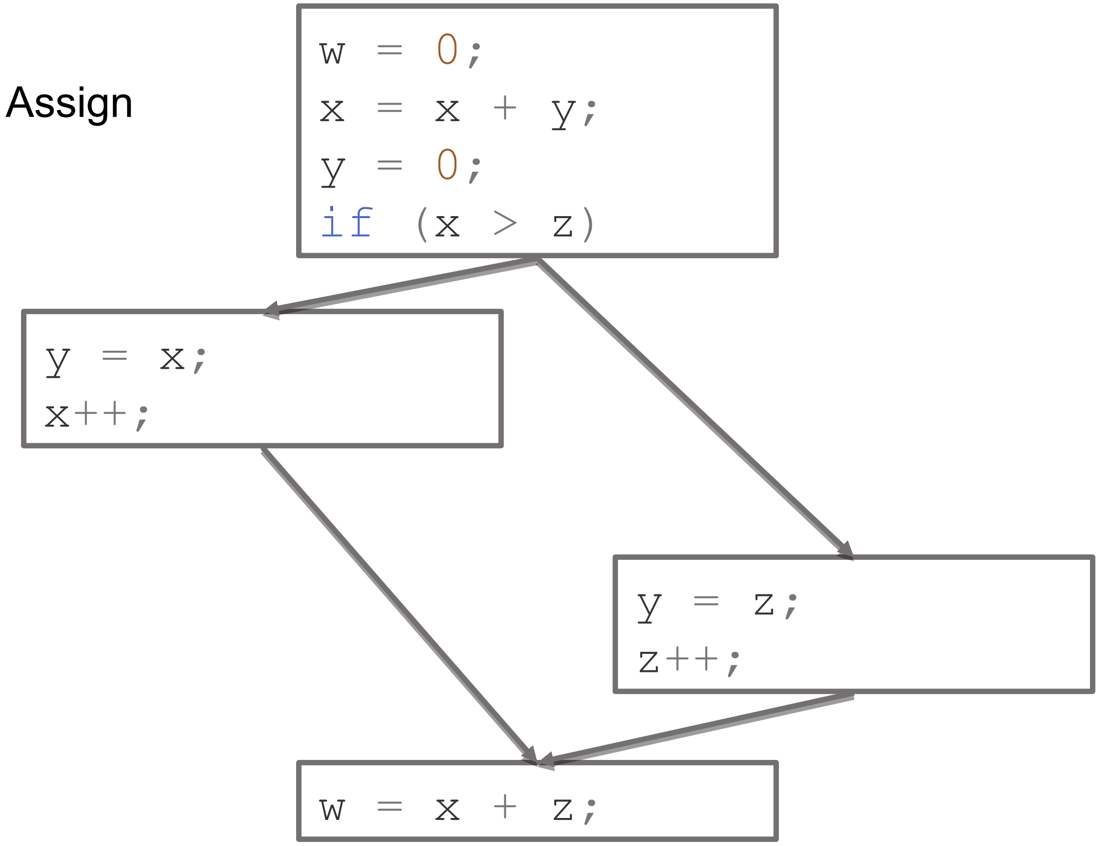
Statement / Assign

```
y = x;
x++;
```

```
y = z;
z++;
```

```
w = x + z;
```

Basic blocks
基本块



```
w = 0;
x = x + y;
y = 0;
if (x > z)
```

```
y = x;
x++;
```

```
y = z;
z++;
```

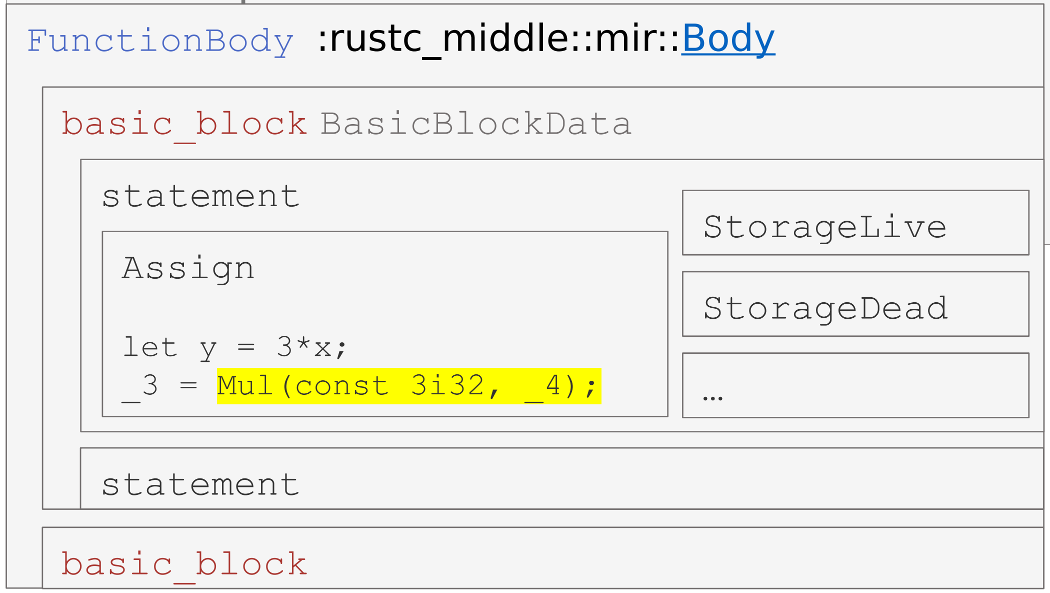
```
w = x + z;
```

Control flow
控制流

RUSTC: API overview (Top-Down)

```
let mir = tcx.optimized_mir(local_def_id.to_def_id());
```

```
for (bb, bbdata: BasicBlockData) in mir.basic_blocks().iter_enumerated() {
  for statement in bbdata.statements.as_slice() {
    if let mir::StatementKind::Assign(assign) = &statement.kind {
      let (_, rval) = assign.as_ref();
      if let mir::Rvalue::Use(operand) = rval {
        if let mir::Operand::Move(place) = operand {
          // …
        }
```


FunctionBody :rustc_middle::mir::Body
basic_block BasicBlockData
statement
Assign
StorageLive
StorageDead
…
statement
basic_block

```
let y = 3*x;
_3 = Mul(const 3i32, _4);
```

```
pub enum Rvalue {
  Use(Operand), // let y = x;
  Repeat(Operand, &Const), // [x; 32]
  Ref(Region, BorrowKind, Place), // &x or &mut x
  AddressOf(Mutability, Place), // &raw const x)
  Cast(CastKind, Operand, Ty),
  BinaryOp(BinOp, Operand, Operand),
  UnaryOp(UnOp, Operand),
// ...
}
```

比较一下 🤔

用来制作代码分析工具

- 运行速度
- 精确性
- 复杂性
- 生态

Exodus

天津 河北

扫一扫上面的二维码图案，加我微信

Thanks

Demo: https://github.com/heymind/rust-china-conf-2020-code-analysis-demo

Both top-down & bottom-up