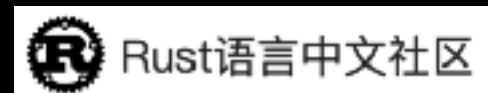


第六课:理解Rust宏

`proc_macro_attribute`、`proc_macro`、`proc_macro_derive`

苏林



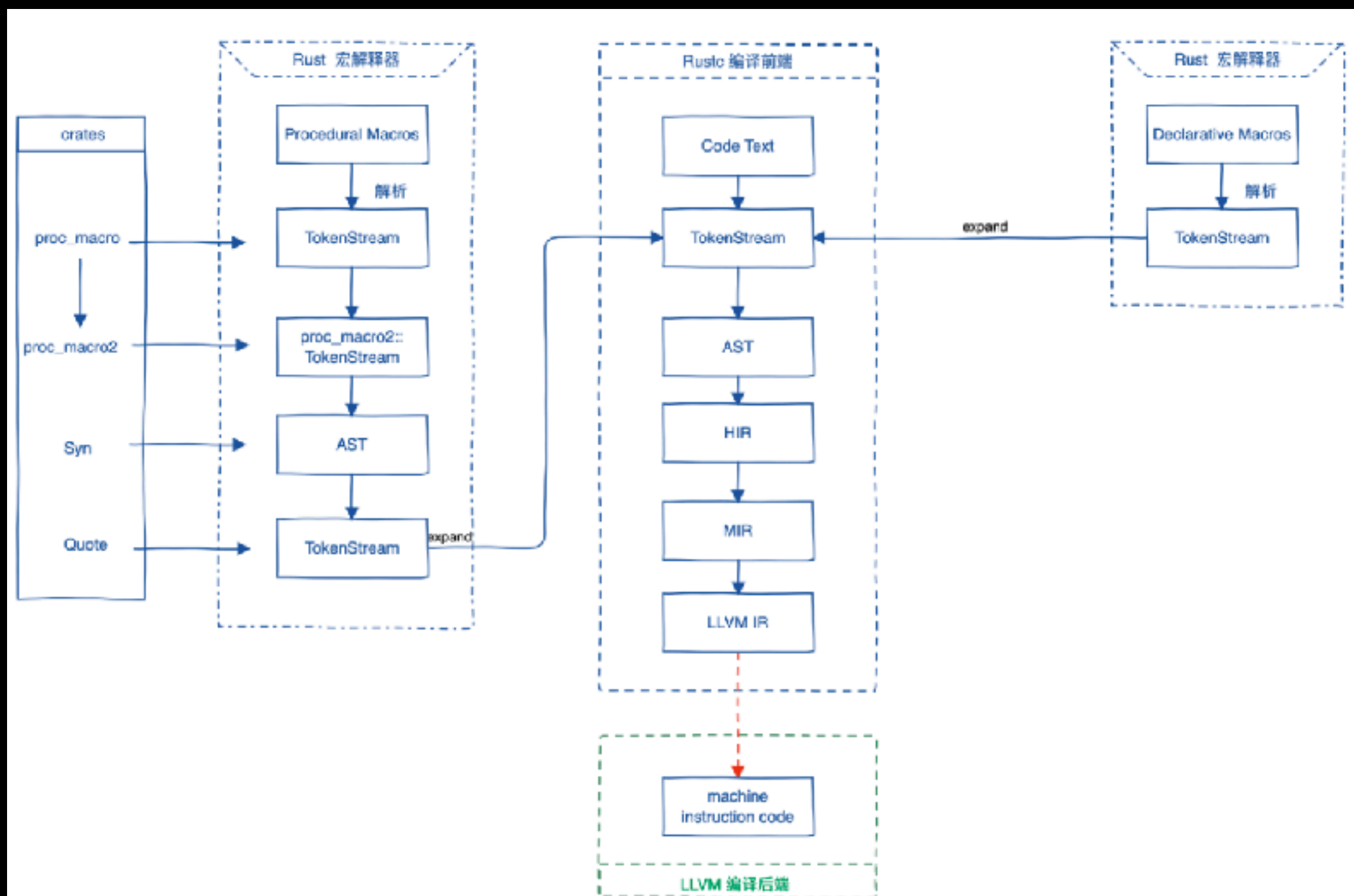
回顾上一次公开课的内容

讲解了如何进行mock测试 和 模糊测试, 通过 `mockall`、`cargo-fuzz`、`afl.rs` 这三个库来进行讲解

今天公开课内容

- 1、结合着Rust编译过程讲述宏生成代码的原理
- 2、通过例子讲解过程宏(派生式、函数式、属性式)

理解Rust编译过程



属性宏 proc_macro_attribute

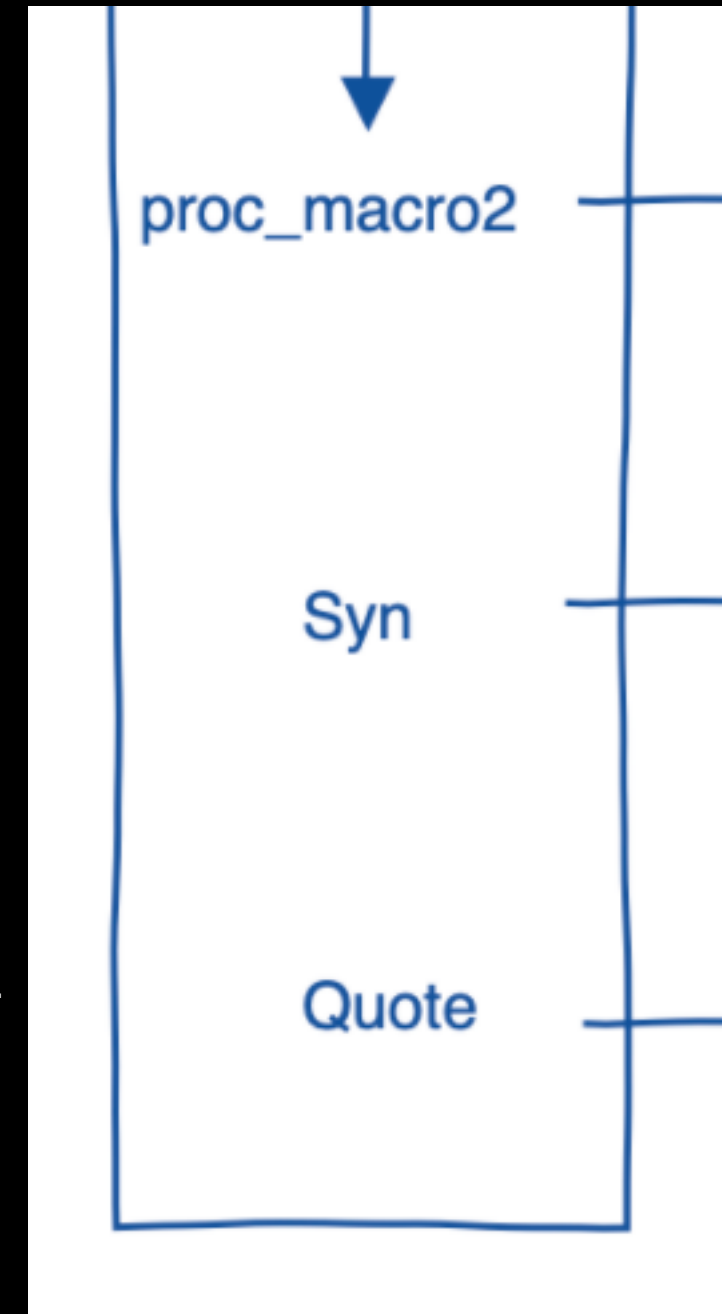
```
1068 #[proc_macro_attribute]
1069 pub fn automock(attrs: proc_macro::TokenStream, input: proc_macro::TokenStream)
1070     -> proc_macro::TokenStream
1071 {
1072     let attrs: proc_macro2::TokenStream = attrs.into();
1073     let input: proc_macro2::TokenStream = input.into();
1074     do_automock(attrs, input).into()
1075 }
```

函数名称, 就是过程宏的名称

[lib]

proc-macro = true

表示这个crate是一个proc-macro过程宏, 这个crate将只能对外导出内部定义的过程宏.



过程宏开发环境搭建

```
sulin@abcd:~/project/rust$ tree rust_macro_study/
rust_macro_study/
├── Cargo.lock
├── Cargo.toml
├── proc_macro_crate
│   ├── Cargo.toml
│   └── src
│       └── lib.rs
└── src
    └── main.rs

3 directories, 5 files
```

```
" Press ? for help

.. (up a dir)
</rust/rust_macro_study/
▼ proc_macro_crate/
  ▼ src/
    lib.rs
    Cargo.toml
  ▼ src/
    main.rs
    Cargo.lock
    Cargo.toml
~
~
~
~
```

```
1 [package]
2 name = "proc_macro_crate"
3 version = "0.1.0"
4 edition = "2021"
5
6 # See more keys and their definitions at https://doc.rust-lang.org/cargo/reference/manifest.html
7
8 [lib]
9 proc-macro = true
10
11 [dependencies]
12 proc-macro2 = "1.0.7"
13 quote = "1"
14 syn = { version = "1.0.56", features = ["full"] }
```

派生式过程宏 `proc_macro_derive`

The screenshot shows a code editor with a dark theme. On the left, a file explorer sidebar lists the project structure: `.. (up a dir)`, `</rust/rust_macro_study/`, `▼ proc_macro_crate/`, `▼ src/`, `lib.rs`, `Cargo.toml`, `▼ src/`, `main.rs`, `Cargo.lock`, and `Cargo.toml`. The main editor area shows the content of `lib.rs`.

```
1 // 属性性过程宏
2 #[proc_macro_attribute]
3 pub fn return_self(_attrs: proc_macro::TokenStream, _input: proc_macro::TokenStream)
4     -> proc_macro::TokenStream
5 {
6     //eprintln!("{:#?}", attrs);
7     //eprintln!("{:#?}", input);
8     proc_macro::TokenStream::new()
9 }
10
11 // 派生式过程宏
12 #[proc_macro_derive(AnswerFn)]
13 pub fn derive_answer_fn(_iten: proc_macro::TokenStream) -> proc_macro::TokenStream {
14     "fn answer() -> i32 {100}".parse().unwrap()
15     //proc_macro::TokenStream::new()
16 }
17
18 // 函数式过程宏
19 #[proc_macro]
20 pub fn make_answer_fn(input: proc_macro::TokenStream) -> proc_macro::TokenStream {
21     "fn answer() -> i32 {200}".parse().unwrap()
22     //proc_macro::TokenStream::new()
23 }
24
25
```

Lines 11 through 16 are enclosed in a red rectangular highlight.

函数式过程宏 proc_macro

[illegible]

QA环节

加群一起交流Rust & Databend

