CTU

CZECH TECHNICAL UNIVERSITY IN PRAGUE

Memory Safety Analysis for Rust GCC

Jakub Dupák

Supervisor: Ing. Pavel Píša PhD.
Project reviewer: MSc. Arthur Cohen

Faculty of Electrical Engineering
Department of Measurement

Borrow Checker Rules

- Move
- Lifetime subset relation
- Borrow must outlive borrowee
- One mutable borrow or multiple immutable borrows
- No modification of immutable borrow data

Borrow Checker Rules

- Move

```
let mut v1 = Vec::new();
v1.push(42)
let mut v2 = v1; // <- Move
println!(v1[0]); // <- Error
```

- Lifetime subset relation
- Borrow must outlive borrowee
- One mutable borrow or multiple immutable borrows
- No modification of immutable borrow data

Borrow Checker Rules

- Move
- **Lifetime subset relation**
- **Borrow must outlive borrowee**

```
fn f() -> &i32 {
  &(1+1)
} // <- Error
```

- **One mutable borrow or multiple immutable borrows**
- **No modification of immutable borrow data**

Borrow Checker Rules

- Move
- Lifetime subset relation
- Borrow must outlive borrowee
- **One mutable borrow or multiple immutable borrows**
- **No modification of immutable borrow data**

```
let mut counter = 0;
let ref1 = &mut counter;
// ...
let ref2 = &mut counter; //  <- Error
```

Checking Functions

```
struct Vec<'a> { ... }

impl<'a> Vec<'a> {
  fn push<'b> where 'b: 'a (&mut self, x: &'b i32) {
    // ...
  }
}
```

Checking Functions

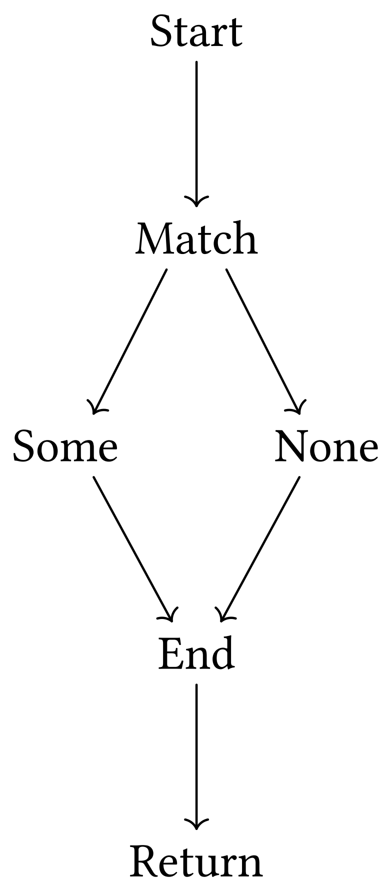
```
struct Vec<'a> { ... }

impl<'a> Vec<'a> {
  fn push<'b> where 'b: 'a (&mut self, x: &'b i32) {
    // ...
  }
}
```

```
let a = 5;                    //   'a    'b    'b: 'a
{                             //
    let mut v = Vec::new();   //    *
    v.push(&a);               //    *     *       OK
    let x = v[0];             //    *     *       OK
 }                            //          *       OK
```

CFG Computation

```
fn f<'a>(map: Map<K, V>) -> &'a V {
  // Lookup key in map.
  // Return reference to value.
  match map.get_mut(&key) {
    Some(value) => value, // Found one.
    None => {
      // Not found.
      // New reference to map!
      map.insert(key, V::default());
    }
  }
}
```


$$\begin{tikzcd}
 & \text{Start} \arrow[d] & \\
 & \text{Match} \arrow[dl] \arrow[dr] & \\
\text{Some} \arrow[dr] & & \text{None} \arrow[dl] \\
 & \text{End} \arrow[d] & \\
 & \text{Return} &
\end{tikzcd}$$

CFG Computation

```rust
fn f<'a>(map: Map<K, V>) -> &'a V {
  // Lookup key in map.
  // Return reference to value.
  match map.get_mut(&key) {
    Some(value) => value, // Found one.
    None => {
      // Not found.
      // New reference to map!
      map.insert(key, V::default());
    }
  }
}
```

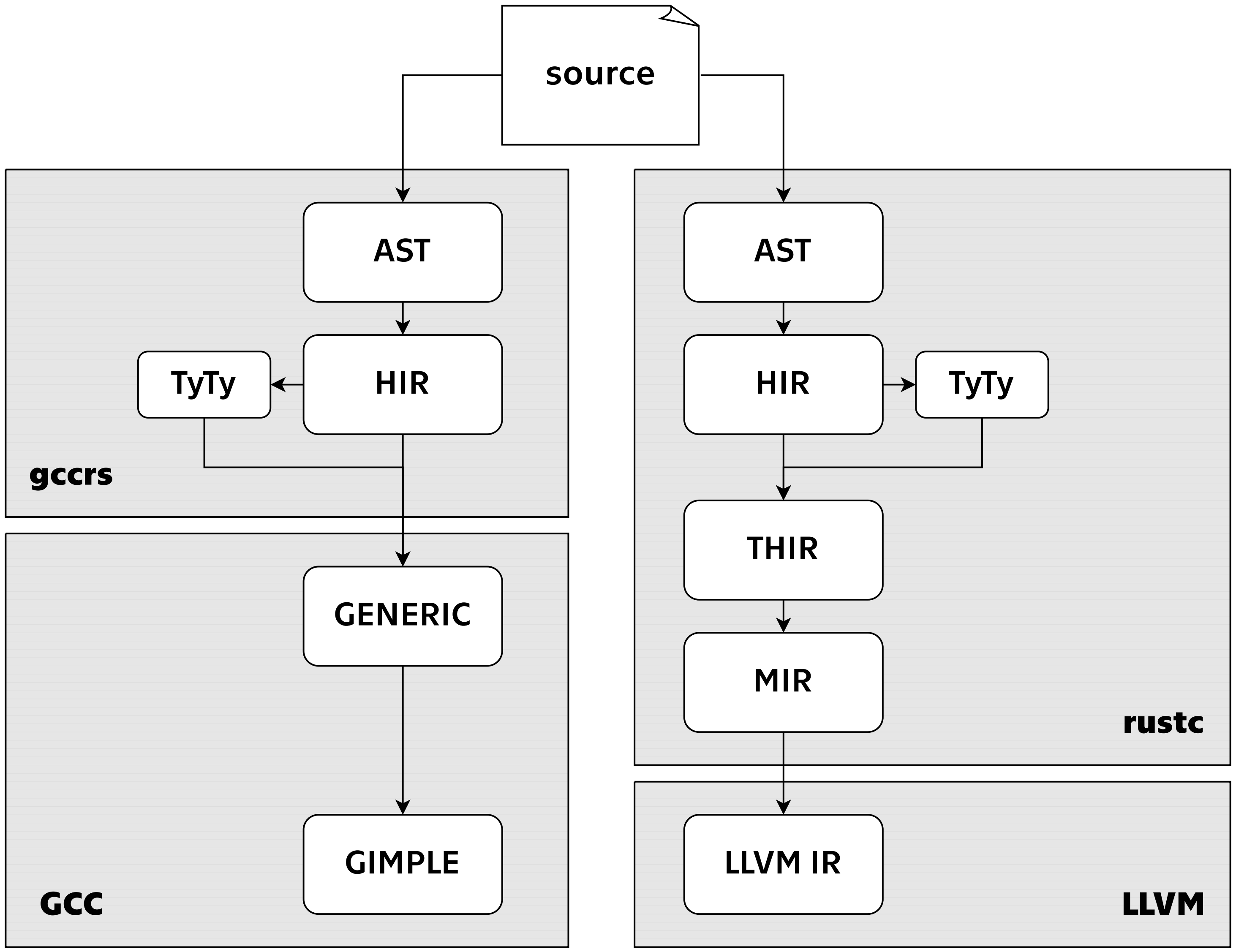
$$\begin{tikzcd}
 & \text{Start} \arrow[d] & \\
 & \text{Match} \arrow[dl] \arrow[dr, dashed] & \\
\text{Some} \arrow[dr] & & \text{None} \arrow[dl] \\
 & \text{End} \arrow[d] & \\
 & \text{Return} &
\end{tikzcd}$$

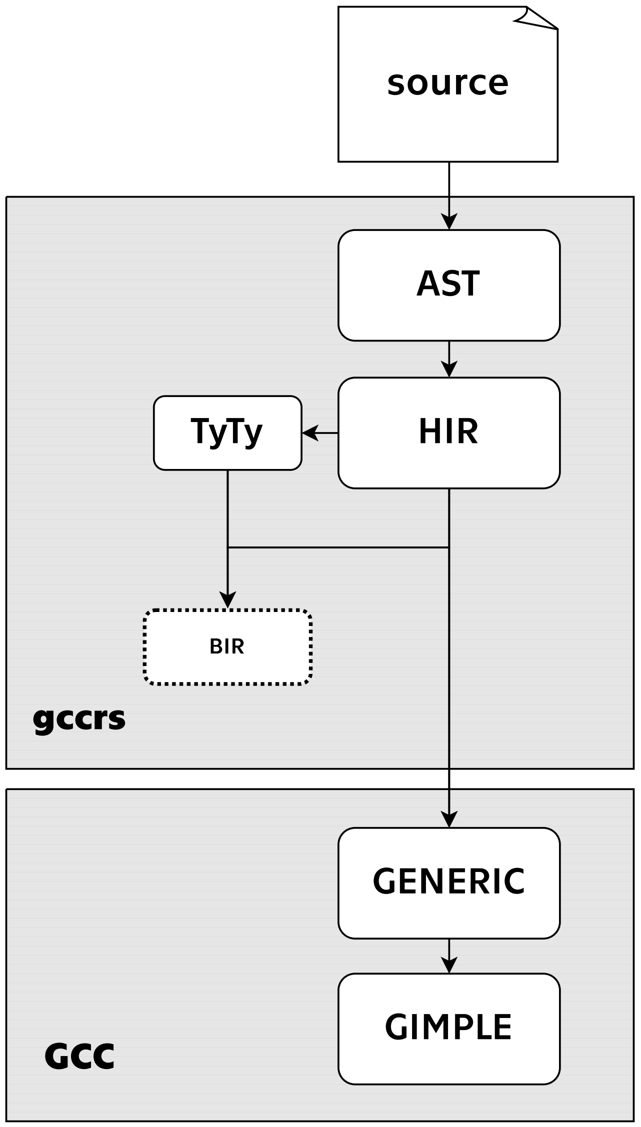
Implementation

- Parsing, AST, HIR
- Lifetime handling in the type checker
- Variance analysis

$$A\langle 'a, T\rangle \leq B\langle 'b, F\rangle \Rightarrow ('a \subseteq 'b) \land (T \leq F)$$

CTU CZECH TECHNICAL UNIVERSITY IN PRAGUE




source
AST
HIR
TyTy
BIR
gccrs
GENERIC
GIMPLE
GCC

Implementation

- Parsing, AST, HIR
- Lifetime handling in the type checker
- Variance analysis
- BIR construction
- Fact collection
- Polonius FFI
- Error reporting

Implementation

- Parsing, AST, HIR
- Lifetime handling in the type checker
- Variance analysis
- BIR construction
- Fact collection
- Polonius FFI
- Error reporting
- Changed +10174 –1374
  - *48%* GCC upstream
  - *11%* Rust GCC
  - *9%* PR in review

Results

- Limitations

- Move errors
- Subset errors
- Access rule errors

Borrow Rules

```
fn mutable_borrow_while_immutable_borrowed() {
    let x = 0;
    let y = &x;      // <---
    let z = &mut x; // <---
    let w = y;
}
```

```
Error: Found loan errors in function
mutable_borrow_while_immutable_borrowed
```

Struct & Method

```
struct Reference<'a> {
    value: &'a i32,
}

impl<'a> Reference<'a> {
    fn new<'a>(value: &'a i32) -> Reference<'a> {
        Reference { value: value }
    }
}
```

Borrow Rules with Struct

```
fn mutable_borrow_while_immutable_borrowed_struct() {
    let x = 0;
    let y = Reference::new(&x);
    let z = &mut x; //~ ERROR
    let w = y;
}
```

```
Error: Found loan errors in function
mutable_borrow_while_immutable_borrowed
```

Subset Rules

```
fn complex_cfg_subset<'a, 'b>(b: bool, x: &'a u32, y:
&'b u32) -> &'a u32 {
    if b {
        y //~ ERROR
    } else {
        x
    }
}
```

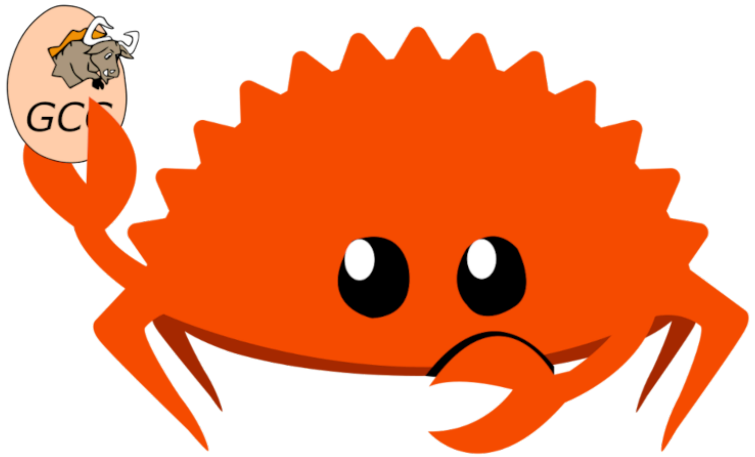
```
Error: Found subset errors in function
complex_cfg_subset
```

CTU
CZECH TECHNICAL UNIVERSITY IN PRAGUE

Future

- Open Source Security support
- GSoC 2024

CTU CZECH TECHNICAL UNIVERSITY IN PRAGUE



Thank You

for your attention